

## Programowanie skryptowe

Dr inż. Stanisław Polak

Grupa Systemów Komputerowych

<https://www.icsr.agh.edu.pl/~polak/>



Notatki

---

---

---

---

---

---

---

---

---

---



## Plan wykładu

- ▶ Język JavaScript
- ▶ Obiektowy model dokumentu HTML
- ▶ Biblioteka React
- ▶ Podstawowe zagadnienia związane z usługą WWW
- ▶ Środowisko uruchomieniowe NodeJS
- ▶ Framework „Express.js”
- ▶ Asynchroniczne zapytania
- ▶ Podstawy języka TypeScript
- ▶ Język Ruby

Notatki

---

---

---

---

---

---

---

---

---

---







## Moduły

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta http-equiv="Content-Type" content="text/html;
   charset=UTF-8">
5   <title>The third JS script</title>
6 </head>
7 <body>
8   <h1>The third JS script</h1>
9   <script type="module">
10     import { welcome } from './functions.js';
11     import './functions.js';
12     // import 'functions.js'; // TypeError: Failed to
13     // resolve module specifier "functions.js".
14     // Relative references must start with either "/"
15     // or "../".
16     console.log(import.meta.url);
17     console.log(document.body.childNodes);
18     welcome(); // Prints "Hello World!"
19     // a = 1; // ReferenceError: a is not defined
20 </script>
21 <script>
22   console.log("Ordinary script executed");
23   console.log(document.body.childNodes);
24   b = 2;
25 </script>
26 <script nomodule>
27   console.log("Modern browsers know both type=module
28   and nomodule, so skip this")
29   console.log("Old browsers ignore script with unknown
30   type=module, but execute this.");
31 </script>
32 </body>
33 </html>

```

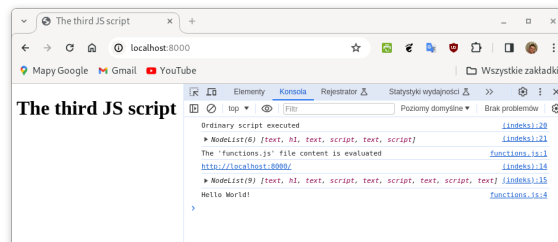
index.html

```

1 console.log("The 'functions.js' file content is evaluated");
2
3 export function welcome() {
4   console.log("Hello World!");
5 }

```

functions.js



Rysunek: Moduły działają tylko poprzez HTTP / HTTPS

```
$ python3 -m http.server
```



Notatki

## Definiowanie zmiennych

```

1 var x = 42;
2 //or
3 let x = 42;
4 //or
5 y = 42; // Not recommended
6 var _y = 42;
7 var $if = 42;
8 var róża = 42;
9 var 1a = 42; // identifier starts immediately after numeric literal
10 console.log(y); // 42
11 console.log(Y); // Y is not defined
12 if = 42; //missing variable name
13 var y = 42
14 console.log(typeof(y)); // number
15 y = "42";
16 console.log(typeof(y)); // string

```

Notatki









## Typ „bigint”

```
1 var x = Number.MAX_SAFE_INTEGER + 1;
2 var y = Number.MAX_SAFE_INTEGER + 2;
3 console.log(x == y); // true
4
5 console.log(BigInt(1)); // 1n
6 console.log(typeof(1n)); // bigint
7
8 // x = BigInt(Number.MAX_SAFE_INTEGER) + 1; // Error. Unable to perform operations on type 'number' and
   // 'bigint'
9 x = BigInt(Number.MAX_SAFE_INTEGER) + 1n;
10 y = BigInt(Number.MAX_SAFE_INTEGER) + 2n;
11 console.log(x == y); //false
```

Notatki

---

---

---

---

---

---

---

---

---

---



## Typ „string”

```
1 var last_name = "Polak";
2 var first_name = 'Stanisław';
3
4 console.log(typeof(last_name)); // string
5 console.log(typeof(first_name)); // string
6
7 console.log("First name=${first_name} Last name=${last_name}"); // First name=${first_name} Last name=${last_name}
8 console.log('First name=${first_name} Last=${last_name}'); // First name=${first_name} Last name=${last_name}
9
10 var a = 11 - "1" ;
11 console.log(a); // 10
12 var b = 11 + "1";
13 console.log(b); // 111
14
15 console.log(typeof(a)); // number
16 console.log(typeof(b)); // string
17
18 last_name[0]='W';
19 console.log(last_name); // "Polak" instead of "Wolak"
20
21 last_name = 'W' + last_name.substr(1);
22 console.log(last_name); // and now "Wolak"
```

Notatki

---

---

---

---

---

---

---

---

---

---





# Typ „string”

## Szablony napisów

### Nieoznakowane

```
1 var a = 2;
2 var str = 'Variable 'a' has value ${a}, 2+2=${2+2}\n';
3 console.log(str);
4 /*****/
5 var str = 'Line 1
6 Line 2';
7 console.log(str);
```

### Oznakowane (otagowane)

```
1 function tag(strings, val1, val2) {
2   result = "String 1:\t\t"+strings[0]+"\n";
3   result += "Raw string 1:\t"+strings.raw[0]+"\n";
4   result += "Value 1:\t\t"+val1+"\n";
5   result += "String 2:\t\t"+strings[1]+"\n";
6   result += "Raw string 2:\t"+strings.raw[1]+"\n";
7   result += "Value 2:\t\t"+val2+"\n";
8   return result;
9 }
10 /*****/
11 var a = 2;
12 var b = 3;
13 str = tag`a+b=\t${a+b}\n, a+b=${a*b}`;
14 console.log(str);
```

Notatki



# Typ „string”

## Sekwencje specjalne

- ▶ \b
- ▶ \f
- ▶ \n
- ▶ \r
- ▶ \t
- ▶ \v
- ▶ \'
- ▶ \"
- ▶ \\
- ▶ \xXX
- ▶ \uXXXX

```
1 console.log("a'\x63'\u0105") // a'c'ą
```

Przykład użycia

Notatki







## Obiekt „Array”

```

1 var tab1 = new Array(1,2,3);//equivalent to: var tab1=Array(1,2,3)
2 var tab2a = new Array(10); //equivalent to: var tab2a = []; tab2a.length=10;
3 var tab2b = new Array("10");
4 var tab3 = [4,'abc',6];
5 var a, rest;
6
7 console.log(tab1.length); // 3
8 console.log(tab2a.length); // 10
9 console.log(tab2b.length); // 1
10 console.log(tab3.length); // 3
11 console.log(tab1[0]); // 1
12 console.log(tab1.0); //missing ) after argument list
13 console.log(tab2a[0]); // undefined
14 console.log(tab2b[0]); // 10
15 console.log(tab3[1]); // abc
16 console.log(tab3[5]); // undefined
17
18 [a, ...rest] = tab3; // Array destructing
19 console.log(a); // 4
20 console.log(rest); // [ 'abc', 6 ]
21
22 var tab4 = new Array(new Array(1,null,undefined),new Array('A','B','C'),new Array('x','y','z'));
23 console.log(tab4.length); // 3
24 console.log(tab4[0]); // 1,
25 console.log(tab4[0][1]); // null
26 var tab1 = new Array(1,2,3);
27 var tab2 = [4,5,6];
28 console.log(tab1); // [1, 2, 3]
29
30 console.log(Array.from("JavaScript")); //[ 'J','a','v','a','S','c','r','i','p','t' ]
31
32 var tab3 = tab1.concat(tab2);
33 console.log(tab3); // 1, 2, 3, 4, 5, 6
34
35 console.log(tab3.splice(1,2)); // 2, 3

```

Notatki



## Obiekt „Map”

```

1 var map = new Map();
2 emptyObject = {};
3 map.set("string","The value associated with the string");
4 map.set(1,{a:10});
5 map.set(emptyObject,1);
6
7 console.log(map); //Map { string: "The value associated with the string", 1: Object, Object: 1 }
8 console.log(map.get(1)); //Object { a: 10 }
9 console.log(map.get(2)); //undefined
10 console.log(map.get("string")); //"The value associated with the string"
11 console.log(map.get({})); //undefined
12 console.log(map.get(emptyObject)); //1
13 console.log(map.size); //3
14 map.delete("string");
15 console.log(map.size); //2
16
17 //Iteration of the hash
18 map.forEach((value,key,map) => {console.log("map["+key+"]="+value)});
19 /*
20 "map[1]=[object Object]"
21 "map[[object Object]]=1"
22 */
23 //Conversion of the array into hash
24 var tab = [{"key1","String"},{"key2",5}];
25 map = new Map(tab);
26 console.log(map); //Map { key1: "String", key2: 5 }

```

Notatki



## Obiekt „RegExp”

```
1 function check(number) {
2   var re = /\d{7}/g; // 'g' - return all matching fragments, not just the first one
3   //lub
4   var re=new RegExp("\\d{7}", "g");
5
6   if(re.test(number))
7     console.log("The correct phone number");
8   else
9     console.log("The telephone number should consist of seven digits");
10 }
11
12 var number1="1234567";
13 var number2="12-34";
14
15 check(number1); // The correct phone number
16 check(number2); // The telephone number should consist of seven digits
```

Sprawdzanie poprawności formatu numeru telefonu

Notatki

---

---

---

---

---

---

---

---

---

---

---

---



## Obiekt „Function”

```
1 var adder = new Function("a", "b", "return a + b");
2 var result = adder(1,2);
3 console.log(result); // 3
4 console.log(adder.length); // 2
5 var obj = {x: 1, y:2};
6 adder = new Function("message", "console.log(message+' ');return this.x + this.y");
7 console.log(adder.call(obj)); // undefined 3
8 console.log(adder.call(obj, 'Value=')); // Value=3
```

Notatki

---

---

---

---

---

---

---

---

---

---

---

---





## Obiekt „navigator”

```

1 //We assume that 'navigator.userAgent' contains the string "Mozilla/4.0 (compatible; MSIE 7.0; Windows
  NT 5.1; .NET CLR 2.0.50727)"
2
3 if (/MSIE (\d+\.\d+)/.test(navigator.userAgent)){ //check if the browser is MSIE x.x;
4   var ieversion=new Number(RegExp.$1) // $1 contains the version number, here: 7.0
5   if (ieversion>=8)
6     document.write("You're using IE8 or above")
7   else if (ieversion>=7)
8     document.write("You're using IE7.x")
9   else if (ieversion>=6)
10    document.write("You're using IE6.x")
11   else if (ieversion>=5)
12    document.write("You're using IE5.x")
13 }
14 else
15   document.write("n/a")

```

Rozpoznawanie wersji przeglądarki IE

Notatki



## Obiekt „window”

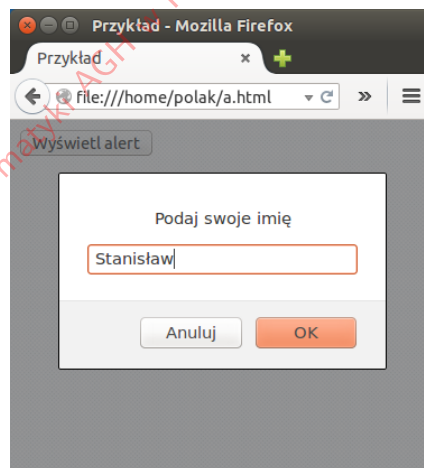
Metody prompt() oraz alert()

```

1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=UTF
  -8">
4 <title>Przykład</title>
5 <script>
6 function load_and_display_the_name(){
7   var first_name=window.prompt('Podaj swoje imię','');
8   window.alert("Witaj "+first_name); //Display welcome text
9 }
10 </script>
11 </head>
12 <body>
13 <form>
14 <button type="button" onClick="load_and_display_the_name();">Wy
 świetl alert</button><br>
15 </form>
16 </body>
17 </html>

```

Otwieranie okna wprowadzania / wyprowadzania danych



Notatki



## Obiekt „window”

Metody `setTimeout()` oraz `clearTimeout()`

```
1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4 <title>An example</title>
5 <script>
6 function delayedAlert(time){
7     timeoutID = window.setTimeout(display, 2000);
8 }
9
10 function display(){
11     window.alert("Hello World!");
12     //timeoutID = window.setTimeout(display, 2000);
13 }
14
15 function stopExecution(){
16     window.clearTimeout(timeoutID);
17 }
18 </script>
19 </head>
20 <body>
21 <form>
22 <button type="button" onClick="delayedAlert();">Show alert</button><br>
23 <button type="button" onClick="stopExecution();">Cancel</button>
24 </form>
25 </body>
26 </html>
```

Po upływie 2 sekund wyświetla się okienko alertu

Notatki



## Obiekt „window”

Metody `setInterval()` oraz `The clearInterval()`

```
1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4 <title>An example</title>
5 <script>
6 function start_cyclic_execution() {
7     timeoutID = window.setInterval(display, 1000);
8 }
9
10 function display(){
11     console.log("Hello World!");
12 }
13
14 function stop_cyclic_execution(){
15     window.clearInterval(timeoutID);
16 }
17 </script>
18 </head>
19 <body>
20 <form>
21 <button type="button" onClick="start_cyclic_execution()">Start</button><br>
22 <button type="button" onClick="stop_cyclic_execution();">Stop</button>
23 </form>
24 </body>
25 </html>
```

Co sekundę wypisywany jest komunikat na konsoli

Notatki





## Obiekt „window”

Metody `requestAnimationFrame()` oraz `cancelAnimationFrame()`

```

1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4 <title>An example</title>
5 <script>
6 function start_cyclic_execution() {
7   requestID=window.requestAnimationFrame(display);
8 }
9
10 function display(){
11   console.log("Hello World!");
12   requestID=window.requestAnimationFrame(display);
13 }
14
15 function stop_cyclic_execution(){
16   window.cancelAnimationFrame(requestID);
17 }
18 </script>
19 </head>
20 <body>
21 <form>
22 <button type="button" onClick="start_cyclic_execution()">Start</button><br>
23 <button type="button" onClick="stop_cyclic_execution()">Stop</button>
24 </form>
25 </body>
26 </html>

```

Cyklicznie, podczas odświeżania ekranu, wypisywany jest komunikat

Notatki



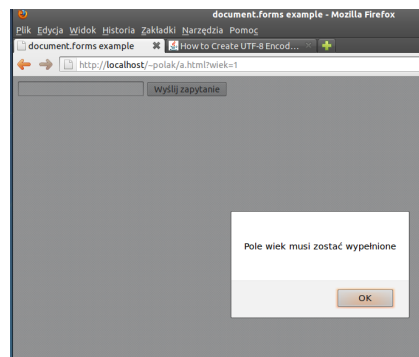
## Obiekt „form”

```

1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4 <title> document.forms example</title>
5 <script>
6 function check() {
7   var form = document.forms[0];
8   //var form = document.forms.form1;
9   //var form = document.forms['form1'];
10  var element = form.elements[0];
11  //var element = form.elements.wiek;
12  //var element = form.elements['wiek'];
13  if (element.value == ""){
14    window.alert("Pole wiek musi zostać wypełnione");
15    return false;
16  }
17  else
18    return true;
19 }
20 </script>
21 </head>
22 <body>
23 <!-- <form ... onSubmit="return false;" -->
24 <form id="form1" action="" onSubmit="return check();">
25   <input name='wiek' type='text'>
26   <input type='submit'>
27 </form>
28 </body>
29 </html>

```

Wstrzymanie wysyłki formularza



Notatki



## Obiekt „Image”

```
1 <html>
2 <head>
3 <script language = "JavaScript">
4
5 function preloader(){
6     bigImage = new Image();
7     bigImage.src = "bigImage.jpg";
8 }
9 </script>
10 </head>
11 <body onLoad="javascript:preloader()">
12 <a href="#" onMouseOver="javascript:document.images[0].src='bigImage.jpg'">
13 <!--
14 lub tak:
15 <a href="#" onMouseOver="javascript:document.img01.src='bigImage.jpg'">
16 -->
17 </a>
18 </body>
19 </html>
```

Wstępne ładowanie dużego obrazka

Notatki



## Obiekt „location”

```
1 <script>
2 if (window.location.protocol == "http:") {
3     var restUrl = window.location.href.substr(5);
4     location.href = "https:" + restUrl;
5 }
6 </script>
```

Przekierowanie do bezpiecznego protokołu HTTPS

Notatki



## Obiekt „history”

## Notatki

---

---

---

---

---

---

---

---

---

---

```
1 history.back(); // equivalent to clicking the 'Back' button
2 history.go(-1); // equivalent to history.back();
```



## Obiekt „screen”

## Notatki

---

---

---

---

---

---

---

---

---

---

```
1 if (screen.pixelDepth < 8) {
2   // use of the "low-color" version of the page
3 } else {
4   // use of the "full-color" version of the page
5 }
```



## Funkcje

- ▶ Definiowanie (inaczej niż w C / Java) — za pomocą słowa kluczowego `function`
- ▶ Wywoływanie funkcji oraz zwracanie wartości przez funkcję — tak samo jak w C / Java
- ▶ Pozwalają definiować prototyp obiektów

```

1 function multiply(a, b=1){
2   var c = a * b;
3   a = 0;
4   b = 0;
5   return c;
6 }
7 function f() { return [1, 2, 3] }
8
9 a = 2;
10 b = 2;
11 var result = multiply(a,b);
12 console.log(result); // 4
13 var result = multiply(a);
14 console.log(result); // 2
15
16 var x, y, z;
17 [x, y, z] = f(); // Returning many values
18 [x, y, z] = (function f() { return [1, 2, 3] })(); //
19 // Simultaneous defining and calling functions
20 console.log(x); // 1
21 //*****
22 const constant = 1;
23 var variable = 2;
24 function constant() {} //Redeclaration of const constant
25 function variable() {}
26 variable(); //variable is not a function

```

Przykładowe funkcje

```

1 function change(x,object1,object2){
2   x = 2;
3   object1.brand = "Fiat";
4   object2 = {brand: "Skoda"};
5 }
6
7 var car1 = {brand: "Ferrari"};
8 var car2 = {brand: "Ferrari"};
9 var variable = 1;
10 console.log(variable); // 1
11 console.log(car1.brand); // Ferrari
12 console.log(car2.brand); // Ferrari
13 change(variable, car1, car2);
14 console.log(variable); // 1
15 console.log(car1.brand); // Fiat
16 console.log(car2.brand); // Ferrari

```

Przekazywanie typów prostych oraz złożonych

Notatki



## Funkcje anonimowe

```

1 //Procedural function
2 function hello1(who) {
3   return 'Hello '+who;
4 }
5 //*****
6 console.log(hello1('world'));// "Hello world"
7 //*****
8 // Function as a variable
9 var hello2 = function (who) {return 'Hello '+who};
10 // or
11 var hello2 = (who) => {return 'Hello '+who};
12 // or
13 var hello2 = (who) => 'Hello '+who;
14 //*****
15 var hello3 = function() {
16   console.log('Hello');
17   console.log('World');
18 }
19 // or
20 var hello3 = () => {
21   console.log('Hello');
22   console.log('World');
23 }
24 console.log(hello2('world'));// "Hello world"
25 hello3(); // "Hello"
26 // "World"

```

```

1 function Person() {
2   // The Person () constructor defines 'this' as an
3   // instance of itself
4   this.age = 0;
5   this.salary = 0;
6 }
7 setInterval(function () {
8   //Here 'this' <=> object 'window' that is, it is
9   //different from 'this' defined in the Person
10  //constructor
11  this.age++;
12  console.log("Age="+this.age);
13 }, 1000);
14 setInterval(() => {
15   this.salary++; //Here 'this' is a Person object
16   console.log("Salary="+this.salary);
17 }, 1000);
18 var person = new Person();

```

Leksykalne this

Notatki





## Zmienne w funkcji

```

1 function fun(x){
2   a = ++x;
3   b = 10;
4 }
5 /*****/
6 a = 0; // <=> var a = 0;
7 console.log(a); // 0
8 console.log(b); //b is not defined
9 fun(a);
10 console.log(a); // 1
11 console.log(b); // 10

```

```

1 function fun(x){
2   var a = ++x;
3   var b = 10;
4 }
5 /*****/
6 a = 0; // <=> var a = 0;
7 console.log(a); // 0
8 console.log(b); //b is not defined
9 fun(a);
10 console.log(a); // 0
11 console.log(b); //b is not defined

```

```

1 let a=1;
2 console.log(a); // 1
3 /*****/
4 for (let i = 0; i<10; i++) {
5   console.log(i);
6   // 1, 2, 3, 4 ... 9
7 }
8 console.log(i); // i is not defined

```

Wyrażenie 'let'

```

1 function fun()
2 {
3   var a = 3;
4   var b = 4;
5   if (a === 3) {
6     let a = 10; // another variable 'a'. Range - interior of the 'if
7               ' block
8     var b = 11; // the same variable 'b' as above. Range - interior
9               of the 'fun' function
10    console.log(a); // 10
11    console.log(b); // 11
12  }
13  console.log(a); // 3
14  console.log(b); // 11
15 }
16 fun();

```

'let' kontra 'var'



## Funkcje ze zmienną liczbą argumentów

```

1 function write() {
2   // go after all arguments
3   for (var i=0; i<arguments.length; i++)
4     console.log(arguments[i]);
5 }
6
7 write("A","B","C");

```

```

1 function write(a, ...others) {
2   console.log(a);
3   console.log(Array.isArray(others));
4   for(counter in others)
5     console.log("others["+counter+"]="+others[
6       counter]);
7   console.log(arguments.length); //SyntaxError: '
8     arguments' object may not be used in
9     conjunction with a rest parameter
10 }
11 write("A","B","C");

```

Notatki





## Definiowanie klas

Notatki

## Składnia JavaScript

```

1 function Human(age, name) {
2   this.age = age;
3   this.name = name;
4 }
5 Human.prototype.getInfo = function () { return 'name=${
6   {this.name} age=${this.age}'; }
7
8 var john = new Human(10, 'John');
9 console.log(john.getInfo()); // name=John age=10
10
11 /*****/
12 function Infant(name) {
13   Human.call(this, 0, name);
14 }
15 Infant.prototype = Object.create(Human.prototype);
16
17 var anna = new Infant('Anna');
18 console.log(anna.getInfo()); // name=Anna age=0

```

## Składnia ES6

```

1 class Human {
2   constructor(age, name) {
3     this.age = age;
4     this.name = name;
5   }
6   getInfo() {
7     return 'name=${this.name} age=${this.age}';
8   }
9 }
10
11 var john = new Human(10, 'John');
12 console.log(john.getInfo()); // name=John age=10
13 /*****/
14 class Infant extends Human {
15   constructor(name) {
16     super(0, name)
17   }
18 }
19
20 var anna = new Infant('Anna');
21 console.log(anna.getInfo()); // name=Anna age=0

```



## Tryb ścisły

Notatki

- ▶ Eliminuje niektóre „pułapki” starszych wersji języka
- ▶ Naprawia błędy, które utrudniały silnikom JavaScriptu dokonywanie optymalizacji
- ▶ Uniemożliwia stosowanie składni, która zostanie wykorzystana w kolejnych wersjach ECMAScriptu

```

1 "use strict";
2 var v = "Cześć! Jestem skrypcem trybu ścisłego!";

```

Włączenie trybu ścisłego dla całego skryptu

```

1 function strict()
2 {
3   // tryb ścisły na poziomie funkcji
4   'use strict';
5   function nested() { return "Ja też"; }
6   return "Cześć! Jestem funkcją trybu ścisłego!" + nested();
7 }
8 function notStrict() { return "A ja nie."; }

```

Włączenie trybu ścisłego dla funkcji











## Wyrażenia tablicowe

```

1 var litery = ["A","B","C"];
2 var liczby = [1,2,3];
3 /******
4 var kwadraty = [for (liczba of liczby) liczba*liczba];
5 console.log(kwadraty); //Wypisze: Array [ 1, 4, 9 ]
6 /******
7 var kwadratyParzystych = [for (liczba of liczby) if(liczba%2 == 0) liczba*liczba ];
8 console.log(kwadratyParzystych); //Wypisze: Array [ 4 ]
9 /******
10 var miks1 = [for (liczba of liczby) for (litera of litery) liczba+litera];
11 console.log(miks1); //Wypisze: Array [ "1A", "1B", "1C", "2A", "2B", "2C", "3A", "3B", "3C" ]
12 /******
13 var miks2 = [for (liczba of liczby) if(liczba > 1) [for (litera of litery) liczba+litera]];
14 console.log(miks2); //Wypisze: Array [ Array[3], Array[3] ]
15 console.log(miks2[0]); //Wypisze: Array [ "2A", "2B", "2C" ]
16 console.log(miks2[1]); //Wypisze: Array [ "3A", "3B", "3C" ]

```

Notatki

---



---



---



---



---



---



---



---



---



---



## Wyrażenia generujące

```

1 var litery = ["A","B","C"];
2 var liczby = [1,2,3];
3 /******
4 var generator = (for (liczba of liczby) liczba*liczba);
5 console.log(generator); //Wypisze: Generator { }
6 console.log(generator.next()); //Wypisze: Object {value: 1, done: false}
7 console.log(generator.next()); //Wypisze: Object {value: 4, done: false}
8 console.log(generator.next()); //Wypisze: Object {value: 9, done: false}
9 console.log(generator.next()); //Wypisze: Object {value: undefined, done: true}
10 var generator = (for (liczba of liczby) liczba*liczba);
11 for(var elem of generator)
12   console.log(elem); /* Wypisze:
13 1
14 4
15 9
16 */
17 /******
18 generator=(for (liczba of liczby) if(liczba > 1) [for (litera of litery) liczba+litera]);
19 console.log(generator.next()); //Wypisze: Object {value: Array[3], done: false}
20 console.log(generator.next()); //Wypisze: Object {value: Array[3], done: false}
21 console.log(generator.next()); //Wypisze: Object {value: undefined, done: true}

```

Notatki

---



---



---



---



---



---



---



---



---



---



## Pośrednicy

```
1 var proceduraObslugi = {
2   deleteProperty(obiektDocelowy, własność){
3     console.log("Kasowanie własności '"+własność+"'");
4     return delete obiektDocelowy[własność]; //Zwraca: true lub false
5     //lub
6     //return Reflect.deleteProperty(obiektDocelowy, własność);
7   }
8 }
9 var p = new Proxy({}, proceduraObslugi);
10 p.a =1;
11 delete p.a; //Wypisze: "Kasowanie własności 'a'"
12 console.log(p.a); //Wypisze: undefined
```

Notatki

---

---

---

---

---

---

---

---

---

---



## Pośrednicy

Przykład zastosowania

```
1 var validator = {
2   set(obiekt, własność, wartość) {
3     if (własność === 'ilość') {
4       if (!Number.isInteger(wartość)) {
5         throw new TypeError('Ilość nie jest liczbą');
6       }
7       if (wartość < 0) {
8         throw new RangeError('Ilość nie może być liczbą ujemną');
9       }
10    }
11  }
12  // Domyślne zachowanie - zapisanie wartości
13  obiekt[własność] = wartość;
14  return true; //true oznacza, że udało się przypisać wartość
15 }
16 };
17
18 var produkt = new Proxy({}, validator);
19 produkt.ilość = 10;
20 console.log(produkt.ilość); //Wypisze: 10
21 produkt.ilość = 'dziesięć'; //TypeError: Ilość nie jest liczbą
22 produkt.ilość = -2; //RangeError: Ilość nie może być liczbą ujemną
```

Notatki

---

---

---

---

---

---

---

---

---

---





## Document Object Model

### Ogólna charakterystyka

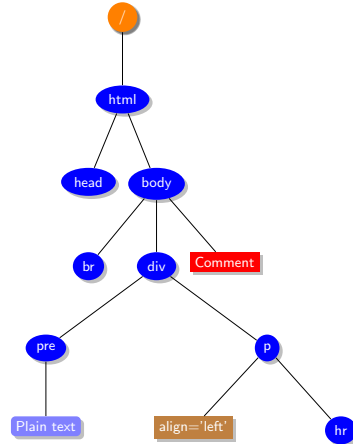
- ▶ Document Object Model (DOM) — obiektowy model dokumentu
- ▶ Dokument — drzewo obiektów
- ▶ Interfejs programowy (API) dla dokumentów HTML i XML
- ▶ Zbiór własności i metod do manipulacji w/w dokumentami

```

1 <!DOCTYPE html>
2 <html>
3 <head></head>
4 <body>
5 <br>
6 <div>
7 <pre>Plain text</pre>
8 <p align='left'>
9 <hr/>
10 </p>
11 </div>
12 <!-- Comment -->
13 </body>
14 </html>
    
```

Dokument HTML

Drzewo DOM



Notatki

---

---

---

---

---

---

---

---

---

---



## Podstawowe własności węzłów

Typ węzła	nodeType	nodeName	nodeValue
Element	Node.ELEMENT_NODE (1)	Nazwa znacznika dużymi literami	null
Atrybut	Node.ATTRIBUTE_NODE (2)	Nazwa atrybutu	Wartość atrybutu
Text	Node.TEXT_NODE (3)	#text	Tekst stanowiący jego treść
Komentarz	Node.COMMENT_NODE (8)	#comment	Treść komentarza
Dokument	Node.DOCUMENT_NODE (9)	#document	null

Notatki

---

---

---

---

---

---

---

---

---

---







## Pobieranie potomków węzłów element oraz tekst

Własność childNodes

Notatki

```

1 <html>
2   <head>
3     <title>An example</title>
4     <script>
5     function start(){
6       var trElements = document.getElementsByTagName("tr");
7       for(let i=0 ; i<trElements.length ; i++){
8         var elements = trElements[i].childNodes; // the type of the 'tdElements' object is "NodeList"
9         for (let j=0;j<elements.length;j++){
10          if (elements[j].nodeType == Node.ELEMENT_NODE) {
11            var string = elements[j].nodeName + ": " + elements[j].childNodes[0].nodeValue;
12            console.log(string);
13          }
14        }
15      }
16    }
17    /* Output:
18    TD: a
19    TD: b
20    TD: c
21    TD: d
22    */
23  </script>
24 </head>
25 <body onLoad="start();" >
26 <table border="1">
27   <tr><td>a</td><td>b</td></tr>
28   <tr><td>c</td><td>d</td></tr>
29 </table>
30 </body>
31 </html>
    
```



## Obsługa atrybutów

Własność attributes oraz metody setAttribute() i removeAttribute()

Notatki

```

1 <html>
2   <head>
3     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4     <title>An examples</title>
5     <script>
6     function change(thickness){
7       var elem1 = document.getElementById("elem1"); // "HTMLTableElement" type object
8       console.log(elem1.getAttribute('border')); // 1
9       console.log(elem1.getAttribute('id')); // elem1
10      elem1.setAttribute('border',thickness);
11    }
12    //You can do it anyway
13    var attributes = elem1.attributes; //obiekt typu "NamedNodeMap" type object
14    console.log(attributes.border.value); // 1
15    console.log(attributes.id.value); // elem1
16    attributes.border.value = thickness;
17  }
18
19  function delete(){
20    var elem1 = document.getElementById("elem1");
21    elem1.removeAttribute('border');
22  }
23 </script>
24 </head>
25 <body>
26 <table border="1" id="elem1">
27   <tr><td>a</td><td>b</td></tr>
28   <tr><td>c</td><td>d</td></tr>
29 </table>
30 <form>
31   <input type="button" value="Change the thickness" onClick="change(2);" >
32   <input type="button" value="Delete" onClick="delete();" >
33 </form>
34 </body>
35 </html>
    
```



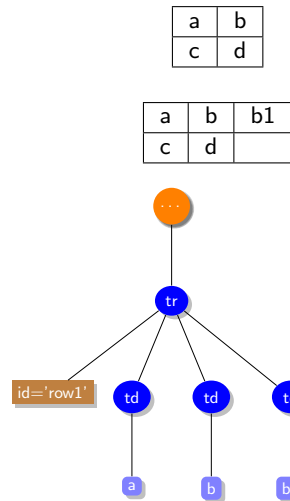
## Wstawienie nowej komórki tabeli na końcu wiersza

Metody `createElement()`, `createTextNode()` oraz `appendChild()`

```

1 <html>
2 <head>
3   <title>An example</title>
4   <script>
5     function insert(){
6       var newTD = document.createElement("td");
7       var newTextNode = document.createTextNode("b1");
8       newTD.appendChild(newTextNode);
9       // newTD.textContent="b1"; // Instead of the above two lines
10      var row1 = document.getElementById("row1");
11      row1.appendChild(newTD);
12    }
13  </script>
14 </head>
15 <body>
16   <table border="1">
17     <tr id="row1"><td>a</td><td>b</td></tr>
18     <tr><td>c</td><td>d</td></tr>
19   </table>
20   <form>
21     <input type="button" value="Insert" onClick="insert();">
22   </form>
23 </body>
24 </html>

```



Notatki

---

---

---

---

---

---

---

---

---

---



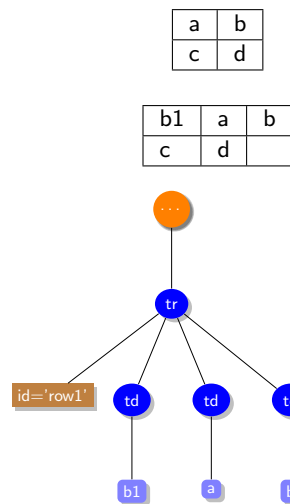
## Wstawienie nowej komórki tabeli na początku wiersza

Metoda `insertBefore()`

```

1 <html>
2 <head>
3   <title>An example</title>
4   <script>
5     function insert(){
6       var newTD = document.createElement("td");
7       var newTextNode = document.createTextNode("b1");
8       newTD.appendChild(newTextNode);
9       var row1 = document.getElementById("row1");
10      var cell1 = row1.getElementsByTagName("td").item(0);
11      row1.insertBefore(newTD, cell1);
12    }
13  </script>
14 </head>
15 <body>
16   <table border="1">
17     <tr id="row1"><td>a</td><td>b</td></tr>
18     <tr><td>c</td><td>d</td></tr>
19   </table>
20   <form>
21     <input type="button" value="Insert" onClick="insert();">
22   </form>
23 </body>
24 </html>

```



Notatki

---

---

---

---

---

---

---

---

---

---







## Ogólna charakterystyka

- ▶ Biblioteka do tworzenia interfejsów graficznych
- ▶ Cechy charakterystyczne:
  - ▶ Wirtualny DOM
  - ▶ JSX

Notatki

---

---

---

---

---

---

---

---

---

---



## Przykład „Hello World” 1

Komponent funkcyjny o nazwie 'Hello'

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <script src="https://unpkg.com/react@latest/umd/react.development.js" crossorigin></script>
5     <script src="https://unpkg.com/react-dom@latest/umd/react-dom.development.js" crossorigin></script>
6     <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
7   </head>
8   <body>
9     <div id="root"></div>
10    <script type="text/babel">
11      // Create a component named 'Hello'
12      function Hello() {
13        return <h1>Hello World!</h1>; // ⇔ React.createElement('h1', null, 'Hello World!');
14      }
15      const container = document.getElementById('root');
16      const root = ReactDOM.createRoot(container);
17      root.render(<Hello />);
18    </script>
19  </body>
20 </html>
```

index.html

Notatki

---

---

---

---

---

---

---

---

---

---



## Przykład „Hello World” 1

Komponent klasowy o nazwie 'Hello'

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <script src="https://unpkg.com/react@latest/umd/react.development.js" crossorigin></script>
5     <script src="https://unpkg.com/react-dom@latest/umd/react-dom.development.js" crossorigin></script>
6     <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
7   </head>
8   <body>
9     <div id="root"></div>
10    <script type="text/babel">
11      // Create a component named 'Hello'
12      class Hello extends React.Component {
13        render() {
14          return <h1>Hello World!</h1>;
15        }
16      }
17      const container = document.getElementById('root');
18      const root = ReactDOM.createRoot(container);
19      root.render(<Hello />);
20    </script>
21  </body>
22 </html>

```

index.html

Notatki



## Przykład „Hello World” 2

```

1 $ npx create-next-app@latest --javascript my-app
2 Would you like to use ESLint? ... No / Yes
3 Would you like to use Tailwind CSS? ... No / Yes
4 Would you like to use 'src/' directory? ... No /
   Yes
5 Would you like to use App Router? (recommended) ...
   No / Yes
6 Would you like to customize the default import
   alias (@/*)? ... No / Yes
7 ...
8 $ cd my-app
9 $ npm run dev

```

```

1 import Hello from "./hello";
2
3 export default function Home() {
4   return (
5     <main className="flex min-h-screen flex-col
6       items-center justify-between p-24">
7       <Hello />
8     </main>
9   );
}

```

src/app/page.js

```

1 'use client'
2
3 import React from "react";
4
5 export default class Hello extends React.Component {
6   render() {
7     return <h1>Hello World!</h1>;
8   }
9 }

```

src/app/hello.js

Notatki



## JSX

Notatki

## Przykład 1

```

1 class Hello extends React.Component {
2   render() {
3     return <div>Hello World 1+2={1+2}</div>;
4   }

```

## Wygenerowany HTML

```
<div>Hello World 1+2=3</div>
```

## Przykład 2

```

1 class Hello extends React.Component {
2   render() {
3     const elements = ["Element 1", "Element 2"];
4     const multiLine = (<ul className='name'>
5       <li>{elements[0]}</li>
6       <li>{elements[1]}</li>
7     </ul>);
8
9     return multiLine;
10  }

```

## Wygenerowany HTML

```

<ul class='name'>
  <li>Element 1</li>
  <li>Element 2</li>
</ul>

```



## Cykl życia komponentu

Notatki

```

1 class Hello extends React.Component {
2
3   constructor(props) {
4     super(props);
5     console.log('constructor()');
6   }
7
8   componentDidMount() {
9     console.log('componentDidMount()');
10  }
11
12  componentWillUnmount() {
13    console.log('componentWillUnmount()');
14  }
15
16  render() {
17    console.log('render()');
18
19    return <div>Hello World</div>;
20  }
21 }
22
23 const container = document.getElementById('root');
24 const root = ReactDOM.createRoot(container);
25 root.render(<Hello />);

```

## Konsola

```

constructor()
render()
componentDidMount()
componentWillUnmount()

```



Pełny wykaz metod można znaleźć na stronie <https://react.dev/reference/react/Component>

## Przekazywanie danych do komponentu

Notatki

```

1 class Hello extends React.Component {
2   render() {
3     return <h1>{this.props.welcome}</h1>;
4   }
5 }
6
7 const container = document.getElementById('root');
8 const root = ReactDOM.createRoot(container);
9 root.render(<Hello welcome="Hello World"/>);

```

## Wygenerowany HTML

```
<h1>Hello World</h1>
```



## Renderowanie warunkowe

Notatki

```

1 class Even extends React.Component {
2   render() {
3     return <div>The number is even</div>
4   }
5 }
6
7 class Odd extends React.Component {
8   render() {
9     return <div>The number is odd</div>
10  }
11 }
12 ...
13 const container = document.getElementById('root');
14 const root = ReactDOM.createRoot(container);
15 root.render(<Number value={Math.floor(Math.random() * 10)} />);

```

```

1 function Number(props) {
2   if (props.value % 2 == 0)
3     return <Even />;
4   else
5     return <Odd />;
6 }

```

Komponent funkcyjny

```

1 class Number extends React.Component {
2   constructor(props) {
3     super(props);
4   }
5
6   render() {
7     if (this.props.value % 2 == 0)
8       return <Even />;
9     else
10      return <Odd />;
11   }
12 }

```

Komponent klasowy







## Przechowywanie danych w komponentcie

```

1 class Hello extends React.Component {
2   constructor(props) {
3     super(props);
4     this.state = { firstName: "John", lastName: "Doe"
5     };
6   }
7
8   clear = () => {
9     this.setState({ firstName: '', lastName: '' })
10  }
11  render() {
12    return (<div><h1>Hello {this.state.firstName} {
13      this.state.lastName} </h1>
14      <button onClick={this.clear}>Clear</button></div>
15    </div>
16  }
17 }
18 const container = document.getElementById('root');
19 const root = ReactDOM.createRoot(container);
20 root.render(<Hello />);

```

Notatki

---



---



---



---



---



---



---



---



---



---



## Obsługa formularza

```

1 class Form extends React.Component {
2   constructor(props) {
3     super(props);
4     this.state = { value: '' };
5   }
6
7   handleChange = (event) => {
8     this.setState({ value: event.target.value });
9   }
10
11  handleSubmit = (event) => {
12    alert('A name was submitted: ' + this.state.value);
13    event.preventDefault();
14  }
15  h
16  render() {
17    return (
18      <form onSubmit={this.handleSubmit}>
19        <label>
20          Name:
21          <input type="text" value={this.state.value} onChange={this.handleChange} />
22        </label>
23        <input type="submit" value="Submit" />
24      </form>
25    );
26  }
27 }
28
29 const container = document.getElementById('root');
30 const root = ReactDOM.createRoot(container);
31 root.render(<Form />);

```

Notatki

---



---



---



---



---



---



---



---



---



---



## Kompozycja komponentów

```

1 class Hello extends React.Component {
2   render() {
3     return (
4       <div>
5         <Welcome text="Hello " />
6         <Welcome text="World" />
7       </div>
8     );
9   }
10 }
11
12 class Welcome extends React.Component {
13   render() {
14     return <span>{this.props.text}</span>
15   }
16 }
17
18 const container = document.getElementById('root');
19 const root = ReactDOM.createRoot(container);
20 root.render(<Hello />);

```

## Wygenerowany HTML

```

<div>
<span>Hello </span>
<span>World</span>
</div>

```

Notatki



## Fragmenty

```

1 class Hello extends React.Component {
2   render() {
3     return (
4       <div>
5         <Welcome text="Hello " />
6         <Welcome text="World" />
7       </div>
8     );
9   }
10 }
11
12 class Welcome extends React.Component {
13   render() {
14     return <span>{this.props.text}</span>
15   }
16 }
17 ...

```

## Wygenerowany HTML

```

<div>
<span>Hello </span>
<span>World</span>
</div>

```

```

1 class Hello extends React.Component {
2   render() {
3     return (
4       <div>
5         <Welcome text="Hello " />
6         <Welcome text="World" />
7       </div>
8     );
9   }
10 }
11
12 class Welcome extends React.Component {
13   render() {
14     return <React.Fragment>{this.props.text}</React.
15       Fragment>
16     // return <>{this.props.text}</>
17   }
18 }
19 ...

```

## Wygenerowany HTML

```

<div>
Hello
World
</div>

```

Notatki



## Wynoszenie stanu w górę

Notatki

```

1 class Echo extends React.Component {
2   constructor(props) {
3     super(props);
4     this.handleChange = this.handleChange.bind(this);
5     this.state = { text: '' };
6   }
7
8   handleChange(newText) {
9     this.setState({ text: newText });
10  }
11
12  render() {
13    return (
14      <React.Fragment>
15        <EchoInput text={this.state.text}
16          handleChange={this.handleChange} />
17        <EchoOutput text={this.state.text} />
18      </React.Fragment>
19    );
20  }
21 }
22 const container = document.getElementById('root');
23 const root = ReactDOM.createRoot(container);
24 root.render(<Echo />);

```

```

1 class EchoOutput extends React.Component {
2   render() {
3     return (
4       <div>Output: {this.props.text}</div>
5     );
6   }
7 }

```

```

1 class EchoInput extends React.Component {
2   constructor(props) {
3     super(props);
4     this.handleChange = this.handleChange.bind(this);
5   }
6
7   handleChange(e) {
8     this.props.handleChange(e.target.value);
9   }
10
11  render() {
12    return (
13      <input value={this.props.text}
14        onChange={this.handleChange} />
15    );
16  }
17 }

```

Źródło: <https://www.geeksforgeeks.org/lifting-state-up-in-reactjs/>



## Tworzenie podstron

Biblioteka 'React Router'

Notatki

```

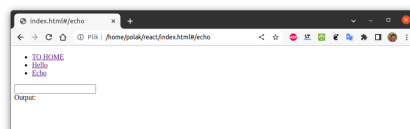
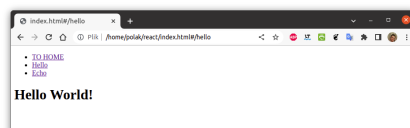
1 ...
2 <head>
3 ...
4 <script src='https://unpkg.com/react-router-dom@5.0.0/umd/react-router-dom.min.js'></script>
5 </head>
6 ...
7 class Route extends React.Component {
8   render() {
9     return (<ReactRouterDOM.HashRouter>
10      <ul>
11        <li><ReactRouterDOM.Link to="/">TO HOME</ReactRouterDOM.Link></li>
12        <li><ReactRouterDOM.Link to="/hello">Hello</ReactRouterDOM.Link></li>
13        <li><ReactRouterDOM.Link to="/echo">Echo</ReactRouterDOM.Link></li>
14      </ul>
15      <ReactRouterDOM.Switch>
16        <ReactRouterDOM.Route path="/echo">
17          <Echo />
18        </ReactRouterDOM.Route>
19        <ReactRouterDOM.Route path="/hello">
20          <Hello />
21        </ReactRouterDOM.Route>
22        <ReactRouterDOM.Route path="/">
23          <Home />
24        </ReactRouterDOM.Route>
25      </ReactRouterDOM.Switch>
26    </ReactRouterDOM.HashRouter>
27  );
28 }
29 }

```

```

30 function Home() {
31   return <h2>Home</h2>;
32 }
33 ...

```





## Wybrane komendy (metody) protokołu HTTP

Notatki

### Komenda „GET”

URL żądania:  
http://www.icsr.agh.edu.pl/index.html

```
1 GET /index.html HTTP/1.1
2 Host: www.icsr.agh.edu.pl
3
```

Żądanie

### Komenda „POST”

URL żądania: http:  
//www.icsr.agh.edu.pl/cgi-bin/search.cgi

```
1 POST /cgi-bin/search.cgi HTTP/1.1
2 Host: www.icsr.agh.edu.pl
3 Content-Length: 46
4
5 query=alpha+complex&casesens=false&cmd=submit
```

Żądanie

```
1 HTTP/1.1 200 OK
2 Date: Mon, 09 Aug 2013 17:02:08 GMT
3 Server: Apache/2.4.4 (UNIX)
4 Content-Length: 1776
5 Content-Type: text/html; charset=utf-8
6
7 <!DOCTYPE html>
8 <html>
9 ...
10 </html>
```

Odpowiedź

```
1 HTTP/1.1 200 OK
2 Date: Mon, 09 Aug 2013 17:02:20 GMT
3 Server: Apache/2.4.4 (UNIX)
4 Content-Length: 1776
5 Content-Type: text/html; charset=utf-8
6 Connection: close
7
8 <!DOCTYPE html>
9 <html>
10 ...
11 </html>
```

Odpowiedź



## Wysyłanie danych z formularza HTML

Notatki

Zatwierdzenie danych → kodowanie → wysyłanie do serwera WWW

```
1 <form method = "... " enctype = "... " action = "... ">
2 ...
3 </form>
```

- ▶ GET
- ▶ POST
- ▶ application/x-www-form-urlencoded
- ▶ multipart/form-data



## Kodowanie „application/x-www-form-urlencoded”

Przykład

Notatki

```

1 <form action="http://www.serwer.com/script">
2 Login: <input name="login" type="TEXT"><br>
3 Password: <input name="password" type="PASSWORD">
4 </form>

```

Dokument HTML

Login: **Stanisław**  
 Password: **Kowalski (Nowak)**

```

1 login=Stanisław&password=Kowalski+(Nowak)

```

Zakodowane dane



## Kodowanie „multipart/form-data”

Przykład

Notatki

```

1 <form action="..." method="POST" enctype="multipart/form-data">
2 <input name="login" type="TEXT">
3 <input name="password" type="PASSWORD">
4 <input name="file" type="FILE" accept="image/jpeg,image/gif">
5 </form>

```

**Stanisław**  
**Kowalski (Nowak)**  
**image.jpg**

```

1 POST /skrypt HTTP/1.0
2 Content-Length: 775
3 Content-Type: multipart/form-data; boundary=-----8152765018186645991017906692
4
5 -----8152765018186645991017906692
6 Content-Disposition: form-data; name="login"
7
8 Stanisław
9 -----8152765018186645991017906692
10 Content-Disposition: form-data; name="password"
11
12 Kowalski (Nowak)
13 -----8152765018186645991017906692
14 Content-Disposition: form-data; name="file"; filename="image.jpg"
15 Content-Type: image/jpeg
16 Content-Transfer-Encoding: binary
17
18 The content of 'image.jpg'
19 -----8152765018186645991017906692

```

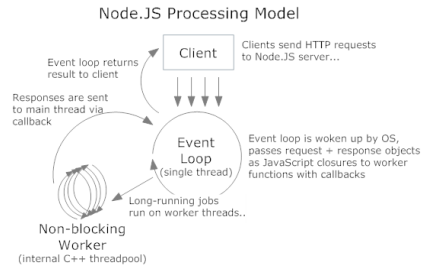


## Node.js

### Ogólna charakterystyka

- ▶ Udostępnia JavaScript po stronie serwera
- ▶ Używa V8 JavaScript Engine
- ▶ System do tworzenia serwisów sieciowych z asynchronicznym WE/WY
- ▶ Wykorzystuje paradygmat programowania sterowanego zdarzeniami
- ▶ Nadaje się dobrze do pisania aplikacji, które wymagają komunikacji w czasie rzeczywistym pomiędzy przeglądarką a serwerem
- ▶ Pojedyncza instancja Node.js działa jako pojedynczy wątek

Pętla zdarzeń — podmiot, który obsługuje / przetwarza zdarzenia zewnętrzne i konwertuje je na wywołania funkcji zwrotnych



Rysunek: Schemat działania pętli zdarzeń w Node.js



## Przykład „Hello World”

```
1 #!/usr/bin/node
2 console.log("Hello World");
```

hello.js

### Notatki







## Moduł

## Tworzenie

## Notatki

```

1 var myModule = require('myModule');
2 //myModule = require("./myModule");
3 /*****
4 console.log(myModule.variable1); //undefined
5 console.log(myModule.variable2); //2
6 console.log(myModule.variable3); //undefined
7 console.log(myModule.variable4); //4
8 console.log(myModule.fun1()); // "fun1"
9 console.log(myModule.fun2()); //Error
10 /*****
11 console.log(myModule); //{ variable2: 2, variable4: 4, fun1: [Function]
12 }
13 console.log(myModule()); //Error

```

script.js

```

1 $ export NODE_PATH='dir1:dir2:...:dirN'

```

```

1 variable1 = 1;
2 exports.variable2 = 2;
3 var variable3 = 3;
4 var variable4 = 4;
5 module.exports.variable4 = variable4;
6 var exports.variable5 = 5; //SyntaxError: Unexpected token .
7 exports.fun1 = function () {
8   return "fun1";
9 };
10 fun2 = function () {
11   return "fun2";
12 };
13 console.log(module);
14 /* Module {
15   ...
16   exports: { variable2: 2, variable4: 4, fun1: [Function] },
17   ... } */
18 exports = function() {
19   return "fun3";
20 }
21 module.exports = function() {
22   return "fun4";
23 }
24 console.log(module.exports); //{ variable2: 2, variable4: 4, fun1:
25   [Function] }
26 console.log(exports); // { variable2: 2, variable4: 4, fun1:
27   [Function] }
28 console.log(module.exports == exports); //true

```

node\_modules/myModule.js



## Moduł

## Korzystanie

## Notatki

## Obsługa pakietów

```

$ npm install nazwa_pakietu
# Moduły ↳ ./node_modules/
# Pliki wykonywalne ↳ ./node_modules/.bin/
# Manuale ↳ nie są instalowane

```

```

$ npm install --global nazwa_pakietu
# Moduły ↳ {prefix}/lib/node_modules/
# Pliki wykonywalne ↳ {prefix}/bin/
# Manuale ↳ {prefix}/share/man/
# {prefix} = np. /usr

```

```

$ npm link nazwa_pakietu
# Wykonuje: ln -s {prefix}/lib/node_modules/nazwa_pakietu/ ./node_modules/

```

```

$ npx nazwa_programu

```



## Obsługa plików

```

1 var fs = require("node:fs");
2
3 fs.writeFileSync('file.txt', '1');
4 console.log('The value 1 has been saved');
5
6 let data = fs.readFileSync('file.txt');
7 console.log("Read value (sync): " + data);
8
9 fs.readFile('file.txt', 'utf-8', function (error,
10   data) {
11   if (error) throw error;
12   console.log("Read value (async): " + data);
13 });
14
15 fs.writeFile('file.txt', '2', function (error) {
16   if (error) throw error;
17   console.log('The value 2 has been saved');
18 });

```

Wersja nieprawidłowa

```

1 var fs = require("node:fs");
2
3 fs.writeFileSync('file.txt', '1');
4 console.log('The value 1 has been saved');
5 let data = fs.readFileSync('file.txt');
6 console.log("Read value (sync): " + data);
7
8 fs.readFile('file.txt', 'utf-8', function (error,
9   data) {
10   if (error) throw error;
11   console.log("Read value (async): " + data);
12
13   fs.writeFile('file.txt', '2', function (error) {
14     if (error) throw error;
15     console.log('The value 2 has been saved');
16   });
17 });

```

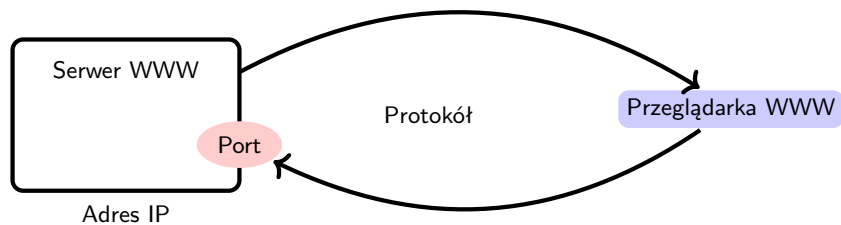
Wersja prawidłowa

Notatki



## Model klient-serwer

### Usługa WWW



Notatki



## Wybrane komendy (metody) protokołu HTTP

Notatki

## Komenda „GET”

URL żądania:

http://www.icsr.agh.edu.pl/index.html

```
1 GET /index.html HTTP/1.1
2 Host: www.icsr.agh.edu.pl
3
```

Żądanie

## Komenda „POST”

URL żądania: http:

//www.icsr.agh.edu.pl/cgi-bin/search.cgi

```
1 POST /cgi-bin/search.cgi HTTP/1.1
2 Host: www.icsr.agh.edu.pl
3 Content-Length: 46
4
5 query=alpha+complex&casesens=false&cmd=submit
```

Żądanie

```
1 HTTP/1.1 200 OK
2 Date: Mon, 09 Aug 2013 17:02:08 GMT
3 Server: Apache/2.4.4 (UNIX)
4 Content-Length: 1776
5 Content-Type: text/html; charset=utf-8
6
7 <!DOCTYPE html>
8 <html>
9 ...
10 </html>
```

Odpowiedź

```
1 HTTP/1.1 200 OK
2 Date: Mon, 09 Aug 2013 17:02:20 GMT
3 Server: Apache/2.4.4 (UNIX)
4 Content-Length: 1776
5 Content-Type: text/html; charset=utf-8
6 Connection: close
7
8 <!DOCTYPE html>
9 <html>
10 ...
11 </html>
```

Odpowiedź



## Wysyłanie danych z formularza HTML

Notatki

Zatwierdzanie danych → kodowanie → wysyłanie do serwera WWW

```
1 <form method = "... " enctype = "... " action = "... ">
2 ...
3 </form>
```

- ▶ GET
- ▶ POST

- ▶ application/x-www-form-urlencoded
- ▶ multipart/form-data



## Kodowanie „application/x-www-form-urlencoded”

Przykład

Notatki

```

1 <form action="http://www.serwer.com/script">
2   Login:   <input name="login" type="TEXT"><br>
3   Password: <input name="password" type="PASSWORD">
4 </form>

```

Dokument HTML

Login: **Stanisław**  
 Password: **Kowalski (Nowak)**

```

1 login=Stanisław&password=Kowalski+(Nowak)

```

Zakodowane dane



## Kodowanie „multipart/form-data”

Przykład

Notatki

```

1 <form action="..." method="POST" enctype="multipart/form-data">
2   <input name="login" type="TEXT">
3   <input name="password" type="PASSWORD">
4   <input name="file" type="FILE" accept="image/jpeg, image/gif">
5 </form>

```

**Stanisław**  
**Kowalski (Nowak)**  
**image.jpg**

```

1 POST /skrypt HTTP/1.0
2 Content-Length: 775
3 Content-Type: multipart/form-data; boundary=-----8152765018186645991017906692
4
5 -----8152765018186645991017906692
6 Content-Disposition: form-data; name="login"
7
8 Stanisław
9 -----8152765018186645991017906692
10 Content-Disposition: form-data; name="password"
11
12 Kowalski (Nowak)
13 -----8152765018186645991017906692
14 Content-Disposition: form-data; name="file"; filename="image.jpg"
15 Content-Type: image/jpeg
16 Content-Transfer-Encoding: binary
17
18 The content of 'image.jpg'
19 -----8152765018186645991017906692

```





## Obsługa formularzy

Obsługa kodowania „application/x-www-form-urlencoded”

```

1 var qs = require('node:querystring');
2 ...
3 function requestListener(request, response) {
4   var url = new URL(request.url, 'http://${request.headers.host}');
5   if(url.pathname == '/') { //generating the form
6     response.writeHead(200, {"Content-Type": "text/html; charset=utf-8"});
7     response.write('<form method="POST" action="/submit">');
8     response.write('<input name="login" value="Jan">');
9     response.write('<input name="password" value="Kowalski (Nowak) aq">');
10    response.write('<input type="submit">');
11    response.write('</form>');
12    response.end();
13  }
14  if(url.pathname == '/submit') { //processing of the form content
15    if(request.method=='GET') {
16      response.writeHead(200, {"Content-Type": "text/plain; charset=utf-8"});
17      response.write(url.searchParams.get('login')+'\n'); //the browser will write: "Jan\n"
18      response.write(url.searchParams.get('password')+'\n'); //the browser will write: "Kowalski (Nowak) aq\n"
19      response.end();
20    }
21    else if(request.method=='POST') {
22      var body='';
23      request.on('data', function (data) {
24        body +=data;
25      });
26      request.on('end', function () {
27        var data = qs.parse(body); //body contains "login=Jan&password=Kowalski+%28Nowak%29+%C4%85%C4%99"
28        response.writeHead(200, {"Content-Type": "text/plain; charset=utf-8"
29          });
30        response.write(data.login+'\n'); //the browser will write: "Jan\n"
31        response.write(data.password+'\n'); //the browser will write: "Kowalski (Nowak) aq\n"
32        response.end();
33      });
34    }
35  }
36 }
37 ...

```

Notatki



Dr inż. Stanisław Polak

113

## Wolnostojący serwer WWW

### Instalacja i uruchamianie

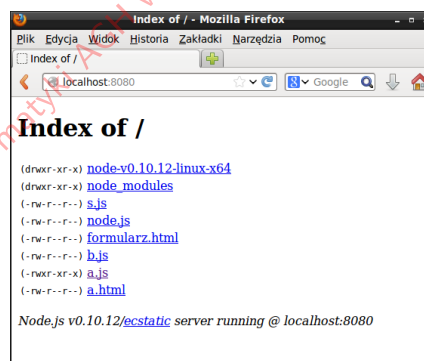
```
$ npm http-server -h
usage: http-server [path] [options]
```

```

options:
  -p          Port to use [8080]
  -a          Address to use [0.0.0.0]
  -d          Show directory listings [true]
  -i          Display autoIndex [true]
  -e --ext    Default file extension if none supplied [none]
  -s --silent Suppress log messages from output
  -h --help  Print this list and exit.
  -c          Set cache time (in seconds). e.g. -c10 for 10
              seconds.
              To disable caching, use -c-1.

```

```
$ npm http-server
Starting up http-server, serving ./ on port: 8080
Hit CTRL-C to stop the server
```



Notatki



Dr inż. Stanisław Polak

114

## Obsługa bazy danych SQLite 3

```

1  var sqlite3 = require('sqlite3');
2
3  var db = new sqlite3.Database(':memory:'); //returns 'Database' object
4
5  db.serialize(function() {
6    db.run("CREATE TABLE products (info TEXT)");
7    var stmt = db.prepare("INSERT INTO products VALUES (?)");//returns 'Statement' object
8    for (var i = 0; i < 2; i++) {
9      stmt.run("Product " + i);
10   }
11   stmt.finalize();
12
13   jsonData = { products: [] };
14   db.each("SELECT rovid AS id, info FROM products", function(err, row) {
15     jsonData.products.push({ id: row.id, info: row.info });
16   }, function () {
17     console.log(JSON.stringify(jsonData)); //JSON.stringify - built-in JS function
18   });
19 });
20 };
21 db.close();

```

bd.js

Notatki



## Tworzenie dodatków (addons) w C++

Przykład „Hello World”

```

1  /*
2  The following program is equivalent to the following JS code:
3  exports.hello = function() { return 'world'; };
4  */
5
6  #include <node.h>
7  using namespace v8;
8
9  void Method(const FunctionCallbackInfo<Value> &args)
10 {
11   Isolate *isolate = args.GetIsolate();
12   args.GetReturnValue().Set(String::NewFromUtf8(
13     isolate, "world")
14     .ToLocalChecked());
15 }
16
17 void init(Local<Object> exports) {
18   NODE_SET_METHOD(exports, "hello", Method); //Associate the name
19   'hello' with the above C++ method and export it
20 }
21 NODE_MODULE(NODE_GYP_MODULE_NAME, init) //there is no semicolon

```

hello.cc

```

1  {
2    "targets": [
3      {
4        "target_name": "hello",
5        "sources": [ "hello.cc" ]
6      }
7    ]
8  }

```

binding.gyp

### Kompilacja i uruchomienie

```

$ npx node-gyp configure
$ npx node-gyp build
$ node hello.js
world

```

```

1  var addon = require('./build/Release/hello');
2  console.log(addon.hello());

```

hello.js

Notatki







## Tworzenie szkieletu aplikacji

Notatki

```
1 $ mkdir MySite
2 $ cd MySite
3 $ vi package.json
4 $ npm install # Installing dependencies
```

```
1 {
2   "name": "MySite",
3   "version": "0.0.1",
4   "private": "true",
5   "dependencies": {
6     "express": "*",
7     "morgan": "*"
8   }
9 }
```

package.json

---

---

---

---

---

---

---

---

---

---



## Plik główny aplikacji

Notatki

```
1 var express = require('express'),
2     logger = require('morgan');
3
4 var app = express();
5
6 // If you uncomment the code snippet below, then
7 // the following function will be executed every
8 // time the app receives a request.
9 /*
10 app.use(function (req, res, next) {
11   console.log('Time:', Date.now());
12   next();
13 });
14 */
15 app.use(logger('dev'));
16
17 app.get('/', function (req, res) {
18   res.send('<h1>Hello World!</h1>');
19 });
20 app.listen(3000);
```

app.js

Terminal 1

```
$ node app
GET / 200 3.137 ms - 21
```

Terminal 2

```
$ curl http://localhost:3000/
<h1>Hello World!</h1>
```

---

---

---

---

---

---

---

---

---

---



## Plik główny aplikacji

## Notatki

```
1 $ npm install #installing dependencies
```

```
1 {
2   ...
3   "dependencies": {
4     ...
5     "pug": "*",
6   }
7 }
```

package.json

---

---

---

---

---

---

---

---

---

---



## Plik główny aplikacji

## Notatki

```
1 var express = require('express'),
2     logger = require('morgan');
3
4 var app = express();
5 var router = express.Router();
6
7 app.set('view engine', 'pug');
8 app.set('views', __dirname + '/views');
9
10 app.use(logger('dev'));
11 app.use(express.static(__dirname + '/public'));
12
13 router.get('/', function (req, res) {
14   res.render('index',
15     { title : 'Przykład' }
16   );
17 });
18
19 app.use('/', router);
20
21 app.listen(3000);
```

app.js

---

---

---

---

---

---

---

---

---

---



## Plik Pug

Notatki

```

1 doctype html
2 html(lang='pl')
3   head
4     title #{title}
5     link(rel='stylesheet', href='/stylesheets/style.
6       css')
7   body
8     header
9       h1 Moja strona
10    main
11      p
12        | Witaj Świecie
13        | Witaj Świecie
14      p
15        | Witaj Świecie
16    aside
17      h1 Nagłówek
18      p
19        | Treść ramki
20    footer
21      p Aplikacja stworzona w oparciu o framework
22        Express

```

views/index.pug

```

1 <!DOCTYPE html>
2 <html lang="pl">
3   <head>
4     <title>Przykład</title>
5     <link rel="stylesheet" href="/stylesheets/style.
6       css">
7   </head>
8   <body>
9     <header>
10      <h1>Moja strona</h1>
11    </header>
12    <main>
13      <p>
14        Witaj Świecie
15        Witaj Świecie
16      </p>
17      <p>Witaj Świecie</p>
18    </main>
19    <aside>
20      <h1>Nagłówek</h1>
21      <p>Treść ramki</p>
22    </aside>
23    <footer>
24      <p>Aplikacja stworzona w oparciu o framework
25        Express</p>
26    </footer>
27  </body>
28 </html>

```



## Plik CSS

Notatki

```

1 aside {
2   float: right;
3   border: 5px solid blue;
4   padding: 1px;
5   margin-bottom: 14px;
6   width: 20%;
7 }
8 main {
9   float: left;
10  background-color: #44f;
11  padding: 5px;
12  width: 75%;
13 }
14 footer {
15   clear: both;
16   border-style: dotted;
17 }
18 header {
19   text-align: center;
20 }

```

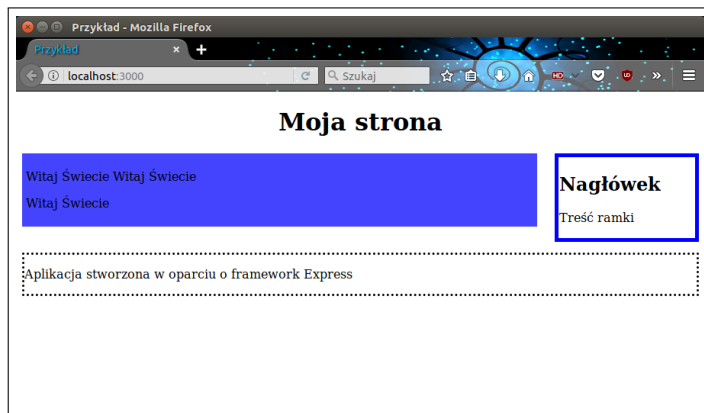
public/stylesheets/style.css



## Uruchamianie aplikacji

Notatki

```
1 $ node app
2 GET / 304 847.588 ms - -
3 GET /stylesheets/style.css 304 4.478 ms - -
```



## Parametry trasy

Notatki

```
1 ...
2 app.get('/students/:studentName/departments/:departmentId', function (req, res) {
3   res.send('studentName=${req.params.studentName}, departmentId=${req.params.departmentId}')
4 });
5 ...
```

app.js

### Terminal 1

```
$ node app.js
GET /students/Jan%20Kowalski/departments/1 200 4.402 ms - 53
```

### Terminal 2

```
$ curl http://localhost:3000/students/Jan%20Kowalski/departments/1
studentName=Jan Kowalski, departmentId=1
```



## Obsługa formularzy

- ▶ <http://localhost:3000/> — strona główna z formularzem
- ▶ <http://localhost:3000/submit> — dane z formularza

Notatki

### Metoda „GET”

```

1 ...
2 router.get('/', function (req, res) {
3   res.send('
4   <form method="GET" action="/submit">
5   <input name="login" value="Jan">
6   <input name="password" value="Kowalski (Nowak) aę"
7   >
8   <input type="submit">
9   </form>
10  ');
11 });
12 router.get('/submit', function (req, res) {
13   res.send('
14   login=${req.query.login}
15   <br>
16   password=${req.query.password}
17   ');
18 });
19 ...

```

app.js

### Metoda „POST”

```

1 ...
2 app.use(express.urlencoded({ extended: false }));
3
4 router.get('/', function (req, res) {
5   res.send('
6   <form method="POST" action="/submit">
7   <input name="login" value="Jan">
8   <input name="password" value="Kowalski (Nowak) aę"
9   >
10  <input type="submit">
11  </form>
12  ');
13 });
14 router.post('/submit', function (req, res) {
15   res.send('
16   login=${req.body.login}
17   <br>
18   password=${req.body.password}
19   ');
20 });
21 ...

```

app.js



## Polecenie express

Notatki

```

1 $ npx express-generator --help
2 Usage: express [options] [dir]
3
4 Options:
5
6   --version      output the version number
7   -e, --ejs      add ejs engine support
8   --pug          add pug engine support
9   --hbs          add handlebars engine support
10  -H, --hogan     add hogan.js engine support
11  -v, --view <engine> add view <engine> support (dust|ejs|hbs|hjs|jade|pug|twig|vash) (defaults to jade)
12
13  --no-view       use static html instead of view engine
14  -c, --css <engine> add stylesheet <engine> support (less|stylus|compass|sass) (defaults to plain css)
15
16  --git           add .gitignore
17  -f, --force     force on non-empty directory
18  -h, --help     output usage information

```



## Generowanie aplikacji „Hello World”

Użycie polecenia express

```
1 $ npx express-generator --view=pug MySite
2 create : MySite/
3 create : MySite/public/
4 create : MySite/public/javascripts/
5 create : MySite/public/images/
6 create : MySite/public/stylesheets/
7 create : MySite/public/stylesheets/style.css
8 create : MySite/routes/
9 create : MySite/routes/index.js
10 create : MySite/routes/users.js
11 create : MySite/views/
12 create : MySite/views/error.pug
13 create : MySite/views/index.pug
14 create : MySite/views/layout.pug
15 create : MySite/app.js
16 create : MySite/package.json
17 create : MySite/bin/
18 create : MySite/bin/www
19
20 change directory:
21 $ cd MySite
22
23 install dependencies:
24 $ npm install
25
26 run the app:
27 $ DEBUG=mysite:* npm start
```

Notatki



## Plik 'package.json'

```
1 {
2   "name": "mysite",
3   "version": "0.0.0",
4   "private": true,
5   "scripts": {
6     "start": "node ./bin/www"
7   },
8   "dependencies": {
9     "cookie-parser": "~1.4.4",
10    "debug": "~2.6.9",
11    "express": "~4.16.1",
12    "http-errors": "~1.6.3",
13    "morgan": "~1.9.1",
14    "pug": "2.0.0-beta11"
15  }
16 }
```

Notatki







### Pozostałe wygenerowane pliki

Notatki

```

1 extends layout.pug
2
3 block content
4   h1= title
5   p Welcome to #{title}

```

views/index.pug

```

1 doctype html
2 html
3   head
4     title= title
5     link(rel='stylesheet', href='/stylesheets/style.
        css')
6   body
7     block content

```

views/layout.pug

```

1 body {
2   padding: 50px;
3   font: 14px "Lucida Grande", Helvetica, Arial, sans
4     -serif;
5 }
6 a {
7   color: #00B7FF;
8 }

```

public/stylesheets/style.css

```

1 $ cd MySite && npm install
2 $ DEBUG=mysite:* npm start
3
4 > mysite00.0.0 start
5 > node ./bin/www
6
7 mysite:server Listening on port 3000 +0ms

```

Instalowanie zależności i uruchamianie aplikacji

---

---

---

---

---

---

---

---

---

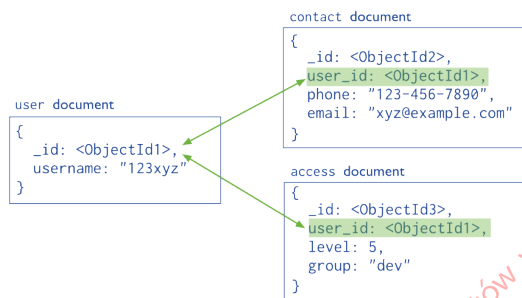
---



### Model danych w MongoDB

Notatki

#### Referencje



#### Osadzone dokumenty

```

{
  _id: <ObjectId1>,
  username: "123xyz",
  contact: {
    phone: "123-456-7890",
    email: "xyz@example.com"
  },
  access: {
    level: 5,
    group: "dev"
  }
}

```

Embedded sub-document

Embedded sub-document

---

---

---

---

---

---

---

---

---

---



Materiały dla studentów Wydziału Informatyki AGH w Krakowie



## Pobieranie i wyświetlanie danych

Notatki

```

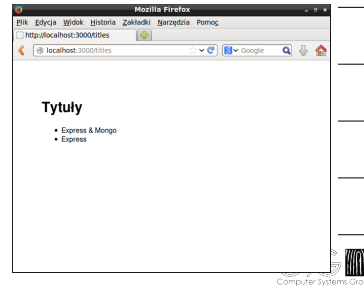
1 ...
2 router.get('/titles', async function (req, res, next) {
3   const MongoClient = require('mongodb').MongoClient;
4   const client = new MongoClient('mongodb://127.0.0.1:27017');
5   await client.connect();
6   const db = client.db('testbase1');
7   const collection = db.collection('datacollection');
8   const docs = await collection.find({}).toArray();
9   res.render('titlelist', {'titlelist': docs});
10  client.close();
11 });
12
13 module.exports = router;
    
```

routes/index.js

```

1 extends layout.pug
2
3 block content
4   hi Tytuły
5   ul
6     each element in titlelist
7       li #{element.title}
    
```

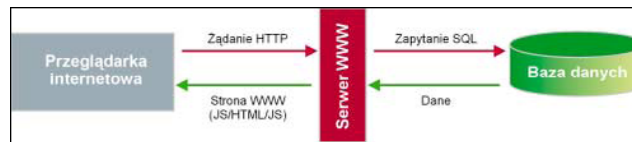
views/titlelist.pug



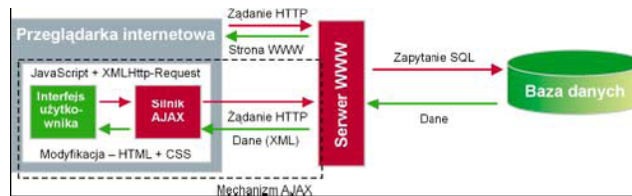
## AJAX

Notatki

- ▶ **AJAX (Asynchroniczny JavaScript i XML)**
- ▶ **AJAX = HTML + CSS + DOM + XMLHttpRequest + XML + JavaScript**
- ▶ **Możliwości:**
  - ▶ Wysyłanie zapytania do serwera bez przeładowywania strony
  - ▶ Aplikacja może dokonywać szybkich, przyrostowych aktualizacji w interfejsie użytkownika bez potrzeby przeładowywania całej strony w przeglądarce
  - ▶ Parsowanie i praca z dokumentami XML
- ▶ Czy warto go zawsze stosować?



Rysunek: Schemat działania typowej witryny internetowej



Rysunek: Schemat działania witryny internetowej opartej o AJAX

## AJAX

Przykład użycia

Notatki

## Zapytanie

```

1 var xhr;
2
3 xhr = new XMLHttpRequest();
4 if (!xhr) {
5   alert('I can not create an XMLHttpRequest object instance
6   ');
7   return;
8 }
9 xhr.onreadystatechange = function() { alertContents(xhr
10 ); };
11 // xhr.onreadystatechange = () => alertContents(xhr);
12 xhr.open('GET', "/requested_file.html", true);
13 xhr.send(null);

```

## Obsługa odpowiedzi

```

1 function alertContents(xhr) {
2   if (xhr.readyState === XMLHttpRequest.DONE) { //
3     XMLHttpRequest.DONE ⇔ 4
4     if (xhr.status === 200) {
5       alert(xhr.responseText);
6       var xmlDoc = xhr.responseXML;
7       var root_node = xmlDoc.getElementsByTagName('
8         td').item(0);
9       alert(root_node.firstChild.data);
10    }
11    else {
12      alert('There was a problem with this task.');
```

```

1 <table>
2 <tr><td>Hello</td></tr>
3 </table>

```

requested\_file.html



Dr inż. Stanisław Polak

142

Asynchroniczne zapytania

AJAX

## AJAX

Wysyłanie danych typu „application/x-www-form-urlencoded”

Notatki

## GET

```

1 ...
2 xhr.open('GET', "/submit.php?field1=value1&field2=value2&...", true);
3 xhr.send(null);

```

## POST

```

1 ...
2 xhr.open('POST', "/submit.php", true);
3 xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
4 xhr.send('field1=value1&field2=value2&...');
```



Dr inż. Stanisław Polak

143

## Odbieranie danych w określonym formacie

```

1 var xhr;
2
3 xhr = new XMLHttpRequest();
4 xhr.responseType = "json";
5 ...
6
7 function alertContents(xhr) {
8     ...
9     alert(xhr.response);
10 }

```

responseType	Typ odbieranych danych
""	DOMString
"text"	DOMString
"arraybuffer"	ArrayBuffer
"blob"	Blob
"document"	Document lub XMLDocument
"json"	JSON

Notatki

---



---



---



---



---



---



---



---



---



---



## Obsługa odpowiedzi za pomocą zdarzeń

```

1 var xhr;
2
3 xhr = new XMLHttpRequest();
4
5 // Called if 'xhr.readyState == XMLHttpRequest.DONE'
6 xhr.addEventListener("load", function (evt) {
7     if (xhr.status === 200) {
8         ...
9     }
10 });
11
12 xhr.addEventListener("error", function (evt) {
13     window.alert('There was a problem with this request.');

```

Notatki

---



---



---



---



---



---



---



---



---



---











## Przykład „Hello World”

```
1 console.log("Hello World");
```

script.ts

```
1 console.log("Hello World");
```

script.js

Notatki

---

---

---

---

---

---

---

---

---

---



## Obsługa błędów

### Błąd typowania

```
1 let alive: boolean = 'abc';  
2 console.log(alive);
```

script.ts

### Błąd składniowy

```
1 let if = 2;  
2 console.log(if);
```

script.ts

```
1 let alive = 'abc';  
2 console.log(alive);
```

script.js

```
1 let ;  
2 if ( = 2)  
3 ;  
4 console.log();  
5 if ()  
6 ;
```

script.js

Notatki

---

---

---

---

---

---

---

---

---

---



## Deklarowanie typu zmiennej

```

1 let alive:boolean = true;
2 let age:number = 48;
3 let name:string = 'Kowalski';
4
5 let names:string[] = ['Jan','Jerzy'];
6 let children_age:Array<number> = [1, 20, 3];
7 let parents_age:Array<number> = [40, "forty
  one"]; //... Type 'string' is not
  assignable to type 'number'.
8
9 let tuple:[string,number,boolean];
10 tuple = ['1',2,true];
11 console.log(tuple[2]); // true
12 tuple = ['1',2,true,4]; // OK
13 tuple = ['1',2]; //... Type '[string, number]'
  is not assignable to type '[string,
  number, boolean]'. ...
14 tuple = [1,2,3]; //... Type '[number, number,
  number]' is not assignable to type '[
  string, number, boolean]'.
15
16 enum Eyes {Blue, Green = 4, Grey, Brown};
17 let eye_color = Eyes.Blue;
18 console.log(Eyes[0]); // Blue
19 console.log(Eyes[4]); // Green
20 console.log(Eyes[5]); // Grey

```

```

21 let anything:any = 4;
22 anything = "String"; //OK
23 anything = true; //OK
24 anything.x; //OK
25 anything(); //OK
26 new anything(); //OK
27 let string: string = anything; //OK
28
29 let something: unknown = 4;
30 something = "String"; //OK
31 something = true; //OK
32 something.x; //... Object is of type 'unknown'
33 something(); //... Object is of type 'unknown'
34 new something(); //... Object is of type '
  unknown'
35 let string2: string = something; // ... Type '
  unknown' is not assignable to type '
  string'.
36
37 function print(message): void {
38   console.log(message);
39 }
40
41 function exception(message): never {
42   throw new Error(message);
43 }
44
45 function loop(): never {
46   while(true){}
47 }

```



Notatki

## Typowanie

```

1 let string:string;
2 string = 1; //Error, you can not assign a number to
  a string type variable
3
4 let number = 1;
5 number = '2'; //Error, you can not assign a string
  to a numeric type variable

```

script.ts

```

1 let string:string;
2 string = <any> 1; //Now it is OK
3 //or
4 string = 1 as any;
5 let number = 1;
6 number = <any> '2'; //Now it is OK
7 //or
8 number = '2' as any; //Now it is OK

```

script.ts

Notatki





# Interfejsy

Typ indeksowany

```
1 interface hashString {
2     [index: string]: string;
3 }
4 let parameters: hashString = {};
5
6 parameters['server'] = 'HP'; // OK
7 let bar:number = parameters['server']; //... error TS2322: Type 'string' is not assignable to type '
   number'.
8 parameters['server'] = 234; //... error TS2322: Type 'number' is not assignable to type 'string'.
```

script.ts

Notatki

---

---

---

---

---

---

---

---

---

---



# Interfejsy

Typy klasowe

```
1 class Person {
2     id:number;
3 }
4
5 interface Validator{
6     checkIdUniqueness(s: number): boolean;
7 }
8
9 class Employee extends Person implements Validator {}
```

script.ts

Notatki

---

---

---

---

---

---

---

---

---

---



























# Ruby w systemie Linux / macOS

Menedżery wersji Ruby

## Ruby Version Manager (RVM)

```
1 $ sudo apt-get update
2 $ sudo apt-get install libgdbm-dev libncurses5-dev automake
   libtool bison libffi-dev
3 $ gpg --keyserver hkp://keys.gnupg.net --recv-keys 409
   B6B1796C275462A1703113804BB82D39DC0E3
4 $ curl -sSL https://get.rvm.io | bash -s stable
5 $ source ~/.rvm/scripts/rvm
6 $ echo "source ~/.rvm/scripts/rvm" >> ~/.bashrc
```

### Instalacja

## Rbenv

```
1 cd
2 git clone https://github.com/rbenv/rbenv.git ~/.rbenv
3 echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bashrc
4 echo 'eval "$(rbenv init -)"' >> ~/.bashrc
5 exec $SHELL
6
7 git clone https://github.com/rbenv/ruby-build.git ~/.rbenv/
   plugins/ruby-build
8 echo 'export PATH="$HOME/.rbenv/plugins/ruby-build/bin:$PATH"'
   >> ~/.bashrc
9 exec $SHELL
```

### Instalacja

## Interpreter Ruby

```
1 $ rvm install <nazwaInterpretera>
2 $ rbenv install <nazwaInterpretera>
```

### Instalacja

```
1 $ rvm list known #wyświetlenie (nazw) dostępnych implementacji
   wraz z numerami ich wersji
2 $ rvm upgrade <nazwaImplementacji> # np. rvm upgrade ruby
3
4 $ rbenv install --list #wyświetlenie (nazw) dostępnych
   implementacji wraz z numerami ich wersji
5 $ rbenv install <nazwaInterpretera>
```

### Aktualizacja

```
1 $ rvm use <nazwaInterpretera> --default #lub rvm use x.x.x --
   default
2 $ rbenv global <nazwaInterpretera>
```

### Określanie domyślnego interpretera

Notatki



# Ruby w systemie Linux / macOS

Ważniejsze implementacje

## Implementacje w języku C (CRuby)

- ▶ MRI (ang. Matz's Ruby Interpreter)
  - ▶ Używany w Ruby  $\leq$  1.8
  - ▶ Tworzone jest drzewo składniowe
  - ▶ Korzysta z „zielonych wątków”
- ▶ YARV (Yet Another Ruby VM)
  - ▶ Używany w Ruby  $\geq$  1.9
  - ▶ Wirtualna maszyna Ruby
  - ▶ Korzysta z wątków natywnych; używa GIL (Global Interpreter Lock)
  - ▶ Oferuje kompilator JIT (Just-in-time) — począwszy od wersji 2.6
  - ▶ Instalacja

```
1 $ rvm install ruby # instalacja najnowszej wersji
   produkcyjnej
2 $ rvm install ruby-head # instalacja najnowszej
   wersji deweloperskiej (testowej)
3 $ rvm install x.x.x # instalacja wersji o numerze
   x.x.x
```

## Inne

- ▶ JRuby
  - ▶ Implementacja w Javie
  - ▶ Używa wątków natywnych, brak GIL
  - ▶ Używa kompilatora JIT
  - ▶ Instalacja

```
1 $ rvm install jruby
```
- ▶ TruffleRuby
  - ▶ Korzysta z:
    - ▶ Truffle — platforma do implementacji wysokowydajnych środowisk językowych korzystających z Javy oraz wirtualnej maszyny Javy
    - ▶ GraalVM — wirtualna maszyna Javy wyposażona w kompilator JIT o nazwie „Graal”
  - ▶ Źródła: <http://www.oracle.com/technetwork/oracle-labs/program-languages/downloads/>

- ▶ Informacje o innych implementacjach — <http://rvm.io/interpreters/>

Notatki





## Ruby w systemie Linux / macOS

### Porównanie wydajności implementacji

```

1 def fibonacci(n)
2   if n < 2
3     return n
4   end
5   return fibonacci(n - 2) + fibonacci(n - 1)
6 end
7 #####
8 def wątki
9   i = 100_000_000
10  2.times.map do
11    Thread.new do
12      50_000_000.times { i -= 1 }
13    end
14  end.each(&:join) #Czekamy aż wszystkie wątki zakończą pracę
15  # Znaczenie '&:nazwa' - patrz slajd nr 234B1okidoc-Start
16  printf("Końcowa wartość i: %d\n", i)
17 end

```

funkcje.rb

```

1 require 'benchmark'
2 load 'funkcje.rb'
3 #####
4 czas_wykonania = Benchmark.realtime do |x|
5   5.times { fibonacci(35) }
6 end
7 printf "Czas wykonania: %f\n", czas_wykonania
8 #####
9 wątki()

```

skrypt.rb

```

1 $ rvm use ruby-2.6.1
2 $ time ruby skrypt.rb
3 Czas wykonania: 10.255780
4 Końcowa wartość i: 0
5
6 real 0m19.201s
7 user 0m19.179s
8 sys 0m0.013s
9 $ time ruby --jit skrypt.rb # Włączona kompilacja JIT
10 Czas wykonania: 3.352901
11 Końcowa wartość i: 0
12
13 real 0m9.019s
14 user 0m17.473s
15 sys 0m0.684s
16 #####
17 $ rvm use 9.2.6.0 # Openjdk wersja "11.0.1"
18 $ time ruby skrypt.rb
19 Czas wykonania: 6.112136
20 Końcowa wartość i: 46414285
21
22 real 0m16.994s
23 user 0m33.553s
24 sys 0m0.313s
25 $ time ruby -X-C skrypt.rb # Wyłączona kompilacja JIT
26 Czas wykonania: 52.742840
27 Końcowa wartość i: 9910369
28
29 real 1m22.684s
30 user 1m56.812s
31 sys 0m0.340s
32 #####
33 $ rvm use truffleruby #GraalVM wersja "1.0.0-rc13"
34 $ time ruby skrypt.rb
35 Czas wykonania: 2.040543
36 Końcowa wartość i: 38358376
37
38 real 0m4.596s
39 user 0m7.361s
40 sys 0m0.157s

```



### Notatki

## RubyGems

- ▶ Standardowy menedżer pakietów (gemów) Rubiego — działa podobnie jak apt, yum, ...
- ▶ Dostępny wraz z Ruby  $\geq 1.9$

### Przykłady użycia

```

1 gem install rails #zainstaluj gem 'Ruby on Rails'
2 gem fetch rails #pobierz gem 'Ruby on Rails', ale go nie instaluj
3 gem uninstall rails #usuń gem 'Ruby on Rails'
4 gem list --local #wyświetl zainstalowane gemy
5 gem list --remote #wyświetl dostępne gemy
6 gem search NAPIŚ --remote #przeszukaj listę dostępnych gemów
7 gem rdoc --all #stwórz dokumentację RDoc/RI dla wszystkich zainstalowanych gemów
8 gem cleanup #usuń stare wersje zainstalowanych gemów

```

### Notatki



## Tworzenie i generowanie dokumentacji

### Program „rdoc”

```

1 # The program takes an initial word or phrase from
2 # the command line (or in the absence of a
3 # parameter from the first line of standard
4 # input). In then reads successive words or
5 # phrases from standard input and reports whether
6 # they are anagrams of the first word.
7 #
8 # Author:: Dave Thomas (mailto:dave@x.y)
9 # Copyright:: Copyright (c) 2002 The Pragmatic Programmers,
10 # LLC
11 # License:: Distributes under the same terms as Ruby
12 #
13 # This class holds the letters in the original
14 # word or phrase. The is_anagram? method allows us
15 # to test if subsequent words or phrases are
16 # anagrams of the original.
17 class Anagram
18 # Remember the letters in the initial word
19 def initialize(text)
20 @initial_letters = letters_of(text)
21 end
22 # Test to see if a new word contains the same
23 # letters as the original
24 def is_anagram?(text)
25 @initial_letters == letters_of(text)
26 end

```

```

27 # Determine the letters in a word or phrase
28 #
29 # * all letters are converted to lower case
30 # * anything not a letter is stripped out
31 # * the letters are converted into an array
32 # * the array is sorted
33 # * the letters are joined back into a string
34 def letters_of(text)
35 text.downcase.delete('`a-z').split('').sort.join
36 end
37 end
38
39 tester = Anagram.new(ARGV.shift || gets)
40 ARGV.each do |text|
41 puts "Anagram!" if tester.is_anagram? text
42 end

```

anagram.rb

```

1 $ rdoc # Generuj pliki '.html'
2 $ rdoc --format=ri # Generuj pliki '.ri'
3 $ rdoc --format=pot # Generuj plik 'rdoc.pot'

```

Generowanie dokumentacji

Notatki



## Wyświetlanie manuala

### Program „ri”

```

1 $ ri --help
2 Usage: ri [options] [names...]
3
4 Where name can be:
5
6 Class | Module | Module::Class
7
8 Class::method | Class#method | Class.method | method
9
10 gem_name: | gem_name:README | gem_name:History
11
12 All class names may be abbreviated to their minimum unambiguous form. If a name
13 is ambiguous, all valid options will be listed.
14
15 A '.' matches either class or instance methods, while #method
16 matches only instance and ::method matches only class methods.
17
18 README and other files may be displayed by prefixing them with the gem name
19 they're contained in. If the gem name is followed by a ':' all files in the
20 gem will be shown. The file name extension may be omitted where it is
21 unambiguous.
22
23 For example:
24
25 ri Fil
26 ri File
27 ri File.new
28 ri zip
29 ri rdoc:README
30
31 Note that shell quoting or escaping may be required for method names containing
32 punctuation:
33
34 ri 'Array.[]'
35 ri compact\!
36
37 To see the default directories ri will search, run:
38
39 ri --list-doc-dirs
40 ...

```

Notatki



## Debugowanie skryptu

```
1 a=1
2 b=2
3 c=3
4 puts a
5 puts b
6 puts c
```

skrypt.rb

```
1 $ gem install byebug
2 $ byebug skrypt.rb
3 [1, 6] in /home/polak/skrypt.rb
4 => 1: a=1
5     2: b=2
6     3: c=3
7     4: puts a
8     5: puts b
9     6: puts c
10 (byebug) display a
11 1: a = nil
12 (byebug) next
13
14 [1, 6] in /home/polak/skrypt.rb
15 1: a=1
16 => 2: b=2
17     3: c=3
18     4: puts a
19     5: puts b
20     6: puts c
21 1: a = 1
```

Debugger „Byebug”

```
1 $ ruby -rdebug skrypt.rb #Załadowanie biblioteki "debug"
2 Debug.rb
3 Emacs support available.
4
5 skrypt.rb:1:a=1
6 (rdb:1) list
7 [-4, 5] in skrypt.rb
8 => 1 a=1
9     2 b=2
10    3 c=3
11    4 puts a
12    5 puts b
13 (rdb:1) display a
14 1: a =
15 (rdb:1) break 3
16 Set breakpoint 1 at skrypt.rb:3
17 (rdb:1) cont
18 Breakpoint 1, toplevel at skrypt.rb:3
19 skrypt.rb:3:c=3
20 1: a = 1
21 (rdb:1) p b
22 2
23 (rdb:1) p c
24 nil
25 (rdb:1) next
26 skrypt.rb:4:puts a
27 1: a = 1
28 (rdb:1) cont
29 1
30 2
31 3
```

Notatki



## Alternatywna konsola Ruby

Program „pry”

### Wybrane cechy

- ▶ Przeglądanie kodu źródłowego
- ▶ Podświetlanie składni
- ▶ Przeglądanie dokumentacji
- ▶ System pomocy, na żywo
- ▶ Integracja powłoki poleceń (uruchamianie edytorów, uruchomienie 'git' oraz 'rake' z poziomu 'pry')
- ▶ Możliwość debugowania kodu

```
1 a=1
2 binding.pry #pierwsza pułapka
3 b=2
4 c=3
5 binding.pry #druga pułapka
6 puts a
7 puts b
8 puts c
```

skrypt.rb

```
1 $ gem install pry #Instalacja 'pry'
2 $ gem install pry-doc #Rozszerzona obsługa dokumentacji dla '
  pry'
3 $ pry
4 [1] pry(main)> puts "Hello World"
5 Hello World
6 => nil
7 [2] pry(main)> exit
8 $ ruby -r pry skrypt.rb #Uruchomienie w trybie „debug”
9 1: a=1
10 => 2: binding.pry #pierwsza pułapka
11     3: b=2
12     4: c=3
13     5: binding.pry #druga pułapka
14     6: puts a
15     7: puts b
16 [1] pry(main)> a
17 => 1
18 [2] pry(main)> b
19 => nil
20 [3] pry(main)> Ctrl+D
21 From: /home/polak/skrypt.rb @ line 5 :
22
23     1: a=1
24     2: binding.pry #pierwsza pułapka
25     3: b=2
26     4: c=3
27 => 5: binding.pry #druga pułapka
28     6: puts a
29     7: puts b
30     8: puts c
31     9:
32
33 [1] pry(main)>
```

Notatki



## Definiowanie zmiennych

```

1 #stała
2 STALA=1
3 =>1
4 STALA=2
5 (irb):2: warning: already initialized constant STALA
6 => 2
7 #definiowanie zmiennych
8 a = 10
9 b = 20
10 c = 30
11 d = 40
12 a, b, c, d = 10, 20, 30, 40
13 #identyfikacja typu
14 d.kind_of? Integer
15 => true
16 d.class
17 => Integer
18 #zmiana typu
19 x = 10
20 => 10
21 x.class
22 => Integer
23 x = "hello"
24 => "hello"
25 x.class
26 => String

```

```

1 y = 20
2 => 20
3 y.to_f
4 => 20.0
5 54321.to_s #kod dziesiątkowy
6 => "54321"
7 54321.to_s(2) #kod dwójkowy
8 => "1101010000110001"
9 54321.to_s(16) #kod szesnastkowy
10 => "d431"
11 54321.to_s(8) #kod ósemkowy
12 => "152061"

```

## Konwersja wartości

Notatki



## Zasięg zmiennych

```

1 x = 10
2 => 10
3 defined? x
4 => "local-variable"
5 $x = 10
6 => 10
7 defined? $x
8 => "global-variable"
9 loopcounter = 10
10 _LoopCounter = 20
11 $welcome = "Welcome to Ruby"
12 gets
13 hello
14 => "hello\n"
15 $.
16 => "hello\n"
17 require 'English'
18 => true
19 $LAST_READ_LINE
20 => "hello\n"
21 $$
22 => 17403
23 @@total = 0
24 @total = 10
25 Stala=1

```

Notatki



## Instrukcja warunkowa if

## Wyrażenie „if”

```

1 if warunek [then]
2   kod...
3 [elsif warunek [then]
4   kod...]]...
5 [else
6   kod...]]
7 end

```

## Składnia

```

1 if 10 < 20 then
2   print "10 jest mniejsze niż 20"
3 end

```

## Przykład użycia

```

1 liczba = 5
2 znak = if liczba > 0
3   'dodatni'
4   elsif liczba < 0
5     'ujemny'
6   else
7     'zero'
8 end
9 znak
10 => "dodatni"

```

## Modyfikator „if”

```
1 kod if warunek
```

## Składnia

```
1 print "10 jest mniejsze niż 20" if 10 < 20
```

## Przykład użycia

Notatki



## Instrukcja warunkowa unless

## Wyrażenie „unless”

```

1 unless warunek [then]
2   kod
3 [else
4   kod ]
5 end

```

## Składnia

```

1 unless str.nil?
2   puts str.length
3 end
4 # W Ruby >= 2.3 powyższe linie można zapisać nastę
5   pująco:
6 # puts str&.length

```

## Przykład użycia

## Modyfikator „unless”

```
1 kod unless warunek
```

## Składnia

```
1 puts str.length unless str.nil?
```

## Przykład użycia

Notatki



## Instrukcja wyboru

Notatki

```

1 ##### Forma 1 #####
2 case wyrażenie
3 [when wyrażenie [, wyrażenie ...] [then]
4   kod ]...
5 [else
6   kod ]
7 end
8
9 ##### Forma 2 #####
10 case
11 [when wyrażenie_logiczne [, wyrażenie_logiczne ...] [then]
12   kod ]...
13 [else
14   kod ]
15 end

```

Składnia

```

1 liczba = 7
2 case liczba
3 when 1,2,3 # Wykonuje: 1 === liczba || 2 === liczba || 3 ===
   liczba
4   "mały"
5 when 4,5,6
6   "średni"
7 when 7,8,9
8   "duży"
9 else
10  "olbrzymi"
11 end
12 => "duży"
13 #####
14 a = 2
15 case
16 when a == 1, a == 2
17   "a ma wartość jeden lub dwa"
18 when a == 3
19   "a ma wartość trzy"
20 else
21   "Nie potrafię określić wartości zmiennej a"
22 end
23 => "a ma wartość jeden lub dwa"
24 #####
25 case url
26 when /http:/
27   "protokół HTTP"
28 ...
29 else
30   "nieznany protokół"
31 end

```

Przykład użycia



## Dopasowywanie wzorców

Notatki

## Definicja

*Dopasowywanie wzorców polega na określeniu wzorców, z którymi powinny się zgadzać niektóre dane, a następnie sprawdzeniu, czy tak się dzieje, i dekonstrukcji danych według tych wzorców.*

Learn You a Haskell for Great Good!  
Miran Lipovaca.

```

1 case [zmienna lub wyrażenie]
2 in wzorzec [if | unless warunek]
3   ...
4 in wzorzec [if | unless warunek]
5   ...
6 else
7   ...
8 end

```

Składnia

```

1 case [0, 1]
2 in[a,b] unless a == b
3   puts "a różne od b"
4 in[a,b] if a == b
5   puts "a równe b"
6 end
7 printf("a=%d b=%d\n",a,b)

```

Przykład

Na wyjściu

a różne od b  
a=0 b=1



## Dopasowywanie wzorców

Operator =&gt;

Notatki

```
1 <wyrażenie> => <wzorzec>
```

Składnia

```
1 config1 = {db: {user: 'admin', password: 'abc123'}}
2 config2 = {db: {password: 'abc123'}}
3 config1 => {db: {user:}}
4 puts "Connect with user '#{user}'"
5 config2 => {db: {user:}}
```

Przykład

Na wyjściu

```
Connect with user 'admin'
script.rb:5:in '<main>': {:db=>{:password=>"abc123"}}: key not found: :user
(NoMatchingPatternKeyError)
```



## Dopasowywanie wzorców

Przykłady

Notatki

### Dopasowywanie tablic

```
1 translation = ['pl', 'Język', 'en', 'Language']
2
3 case translation
4   in ['pl', orig_text, 'fr', trans_text]
5     puts "Tłumaczenie francuskie:\n\t#{orig_text} =>
6       #{trans_text}"
7   in ['pl', orig_text, 'en', trans_text]
8     # To zostanie wykonane
9     puts "Tłumaczenie angielskie:\n\t#{orig_text} =>
10    #{trans_text}"
11 end
```

### Dopasowywanie haszy

```
1 translation = {orig_lang: 'pl', trans_lang: 'en',
2               orig_txt: 'Język', trans_txt: 'Language' }
3
4 case translation
5   in {orig_lang: 'pl', trans_lang: 'en', orig_txt:
6     orig_txt, trans_txt: trans_txt}
7     puts "#{orig_txt} => #{trans_txt}"
8 end
```

Na wyjściu

```
Tłumaczenie angielskie:
Język => Language
```

Na wyjściu

```
Język => Language
```







## Instrukcje break, redo oraz next

```
1 i = 0
2 while p "Wyrażenie 'while'"
3   i += 1
4   printf "\ti=%d\n", i
5   case i
6   when 2
7     puts "\tWykonuję 'next'"
8     next
9   when 4
10    puts "\tWykonuję 'redo'"
11    redo
12  when 6
13    puts "\tWykonuję 'break'"
14    break
15  end
16  puts "\tJestem na końcu 'while'"
17 end
18 puts 'Koniec'
```

Notatki

---

---

---

---

---

---

---

---

---

---

---

---



## Pętla loop

```
1 #Pętla nieskończona
2 loop do
3   puts "Witaj"
4 end
5
6 #Pętla skończona
7 i = 1
8 loop do
9   puts i
10  i += 1
11  j = 2
12  break if i > 3
13 end
14 puts j #undefined local variable or method 'j'
```

Notatki

---

---

---

---

---

---

---

---

---

---

---

---







## Przedziały

### Metody

```
1 words = 'cab'..'car'
2 words.min
3 => "cab"
4 words.max
5 => "car"
6 words.include?('can')
7 => true
8 words.reject {|subrange| subrange < 'cal'}
9 => ["cal", "cam", "can", "cao", "cap", "caq", "car"]
10 words.each {|word| puts "Hello " + word}
11 Hello cab
12 Hello cac
13 ...
14 Hello car
```

### Notatki



## Przedziały

### Przykłady zastosowania

```
1 while input = gets
2   puts input + " triggered" if input =~ /start/..
3   input =~ /end/
4 end
```

```
1 score = 70
2 result =
3   case score
4     when 0..40: "Nie zaliczył (oblał test)"
5     when 41..60: "Zaliczył"
6     when 61..70: "Zaliczył z wysokim wynikiem"
7     when 71..100: "Zaliczył z wyróżnieniem"
8     else "Nieprawidłowy wynik testu"
9   end
10 puts result
```

### Notatki



## Tablice zwykłe

```

1 days_of_week = Array.new(7)
2 => [nil, nil, nil, nil, nil, nil, nil]
3 days_of_week = Array.new(7, "today")
4 => ["today", "today", "today", "today", "today", "today", "today"]
5 days_of_week = Array[ "Mon", "Tues", "Wed", "Thu", "Fri", "Sat", "Sun" ]
6 #lub
7 days_of_week = [ "Mon", "Tues", "Wed", "Thu", "Fri", "Sat", "Sun" ]
8 => ["Mon", "Tues", "Wed", "Thu", "Fri", "Sat", "Sun"]
9 days_of_week.empty?
10 => false
11 days_of_week.size
12 => 7
13 days_of_week.length
14 => 7
15 days_of_week[0]
16 => "Mon"
17 days_of_week.at(0)
18 => "Mon"
19 days_of_week.first
20 => "Mon"
21 days_of_week[0].class
22 => String
23 days_of_week.index("Wed")
24 => 2
25 days_of_week[1, 3]
26 => ["Tues", "Wed", "Thu"]
27 days_of_week[1..3]
28 => ["Tues", "Wed", "Thu"]

```

Notatki



## Tablice zwykłe c.d.

```

1 matrix = [[1,2,3],[3,4,5],[5,6,7]]
2 matrix[1]
3 => [3, 4, 5]
4 matrix[1][1]
5 => 4
6 matrix[1,1]
7 => [3, 4, 5]]
8 #####
9 operating_systems = ["Fedora", "SuSE", "RHEL", "Windows", "
   macOS"]
10 linux_systems = ["RHEL", "SuSE", "PCLinuxOS", "Ubuntu", "
   Fedora"]
11 operating_systems | linux_systems
12 => ["Fedora", "SuSE", "RHEL", "Windows", "macOS", "PCLinuxOS",
   "Ubuntu"]
13 operating_systems & linux_systems
14 => ["Fedora", "SuSE", "RHEL"]
15 operating_systems - linux_systems
16 => ["Windows", "macOS"]
17 days1 = ["Mon", "Tue", "Wed"]
18 days2 = ["Thu", "Fri", "Sat", "Sun"]
19 days = days1 + days2
20 days
21 => ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"]
22 days1
23 => ["Mon", "Tue", "Wed"]
24 #####
25 days1 = ["Mon", "Tue", "Wed"]
26 days2 = ["Thu", "Fri", "Sat", "Sun"]
27 days = days1.concat(days2)
28 => ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"]
29 days
30 => ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"]
31 days1
32 => ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"]
33
34 #####
35 days1 = ["Mon", "Tue", "Wed"]
36 days2 = ["Thu", "Fri", "Sat", "Sun"]
37 days = days1 << days2
38 days
39 => ["Mon", "Tue", "Wed", ["Thu", "Fri", "Sat", "Sun"]]
40 days1
41 => ["Mon", "Tue", "Wed", ["Thu", "Fri", "Sat", "Sun"]]

```

Notatki



## Tablice asocjacyjne

Notatki

```

1 hash = { "jeden" => 1, "dwa" => 2, "trzy" => 3 }
2 hash["jeden"]
3 => 1
4 hash1 = {jeden:1, dwa:2, trzy:3} #Składnia dostępna począwszy od wersji 1.9
5 => {:jeden=>1, :dwa=>2, :trzy=>3}
6 hash1[:jeden]
7 => 1
8 histogram = Hash.new(0)
9 => {}
10 histogram["Ala"]
11 => 0
12 histogram["Ala"] += 1
13 => 1
14 histogram["Ala"]
15 => 1
16 hash.has_key?("jeden")
17 => true
18 hash.has_value?("jeden")
19 => false

```



## Uporządkowane tablice asocjacyjne

Notatki

```

1 h = {"c" => 3, "a" => 1, "b" => 2}
2 h.each do |k, v|
3   p [k, v]
4 end
5
6 #Wynik działania:
7 #   Ruby 1.9+
8 ["c", 3]           ["a", 1]
9 ["a", 1]           ["b", 2]
10 ["b", 2]           ["c", 3]

```

```

1 h = {}
2 h["a"] = "foo"
3 h["b"] = "bar"
4 p h
5 # Na wyjściu:
6 {"a"=>"foo", "b"=>"bar"}
7
8 h["a"] = nil
9 p h
10 #Na wyjściu:
11 {"a"=>nil, "b"=>"bar"}

```



## Łańcuchy znaków

### Tworzenie i formatowanie

```

1 ##### tworzenie łańcucha #####
2 myString = String.new
3 => ""
4 myString = String.new("This is my string. Get your own string")
5 myString = String("This is also my string") #metoda 'String'
   jest metodą modułu "Kernel" - konwersja do typu
   napisowego
6 myString = "This is also my string"
7 myString = 'This is also my string'
8 ##### wypisanie łańcucha #####
9 '#{myString} #{1+3}' lub %Q#{myString} #{1+3}& lub %&#{
   myString} #{1+3}& lub %[#{myString} #{1+3}]
10 => "This is also my string 4"
11 '#{myString}' lub %q#{myString}& lub %q#{myString}
12 => "\#{myString}"
13 'ls' lub %xls& lub %x{ls}
14 => Wynik wykonania komendy 'ls'
15 ##### składnia HereDocs #####
16 myText = <<DOC
17 1
18 #{myString}
19 DOC
20 DOC
21 => "1\nThis is also my string\n      DOC\n"
22 myText = <<-DOC
23 1
24 #{myString}
25 DOC
26 => "1\nThis is also my string\n"

```

```

27 ##### Tworzenie tablicy napisów #####
28 %w(1 2 aaa)
29 => ["1", "2", "aaa"]
30 ##### Formatowanie łańcucha #####
31 a=1
32 b=2
33 "a=%d" % a
34 => "a=1"
35 "a=%d, b=%d" % [a,b]
36 => "a=1, b=2"
37 "c=%{c}, d=%{d}" % {:c=>3, :d=>4}
38 => "c=3, d=4"
39 ##### Wybrane metody #####
40 myString = "Hello"
41 myString.length
42 => 5
43 myString.size
44 => 5
45 myString.empty?
46 => false
47 myString.sub('H','h')
48 => "hello"
49 myString
50 => "Hello"
51 myString.sub!('H','h')
52 => "hello"
53 myString
54 => "hello"
55 myString.object_id
56 => 9565120
57 myString.sub!('h','H')
58 => "Hello"
59 myString.object_id
60 => 9565120

```

### Notatki



## Łańcuchy znaków

### Łączenie, dostęp do fragmentów oraz porównywanie

```

1 ##### Łączenie #####
2 myString = "Welcome " + "to " + "Ruby!" #konkatenacja (nie modyfikuje łańcucha z lewej)
3 => "Welcome to Ruby!"
4 myString = "Welcome " "to " "Ruby!" #jw.
5 => "Welcome to Ruby!"
6 myString = "Welcome ".concat("to ").concat("Ruby!") #dołącza do łańcucha z lewej (modyfikuje go)
7 => "Welcome to Ruby!"
8 myString = "Welcome " << "to " << "Ruby!" #jw.
9 => "Welcome to Ruby!"
10 ##### dostęp do fragmentów #####
11 myString["Ruby"]
12 => "Ruby"
13 myString["Perl"]
14 => nil
15 myString[3] #Ruby 1.9+
16 => "c"
17 myString[3].ord #Ruby 1.9+
18 => 99
19 myString[3].chr #Ruby 1.8
20 => "c"
21 myString[11, 4]
22 => "Ruby"
23 myString[0..6]
24 => "Welcome"
25 myString[/[A-Z]/]
26 => "W"
27 myString.index('Ruby')
28 => 11

```

### Notatki



## Łańcuchy znaków

Zastępowanie, substytucja, wstawianie

```
1 ##### podmiana #####
2 myString = "Welcome to JavaScript!"
3 myString["JavaScript"]= "Ruby"
4 puts myString
5 => "Welcome to Ruby!"
6 myString = "Welcome to JavaScript!"
7 myString[10] = "Ruby"
8 puts myString
9 => "Welcome toRubyJavaScript!"
10 myString = "Welcome to JavaScript!"
11 myString[8..20] = "Ruby"
12 puts myString
13 => "Welcome Ruby!"
14 ##### substytucja #####
15 myString = "Welcome to PHP"
16 myString.gsub("PHP", "Ruby")
17 => "Welcome to Ruby"
18 myString.replace("Goodbye to PHP!")
19 => "Goodbye to PHP!"
20 ##### powielanie #####
21 myString * 3
22 => "Goodbye to PHP!Goodbye to PHP!Goodbye to PHP!"
23 ##### wstawianie #####
24 myString1 = "Paris in Spring"
25 myString2 = "Paris in Spring"
26 myString1[8]=" the" # podmiana, a nie wstawienie przed
27 => " the"
28 myString1
29 => "Paris in theSpring"
30 myString2.insert(8, " the")
31 => "Paris in the Spring"
32 myString2
33 => "Paris in the Spring"
```

Notatki



## Łańcuchy znaków

Konwersja

```
1 myArray = Array("ABCDEFGHIJKLMNOP")
2 => ["ABCDEFGHIJKLMNOP"]
3 myArray = "ABCDEFGHIJKLMNOP".split
4 => ["ABCDEFGHIJKLMNOP"]
5 myArray = "ABCDEFGHIJKLMNOP".split(//)
6 => ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P"]
7 "1000".to_i
8 => 1000
9 Integer("aaa")
10 #ArgumentError: invalid value for Integer(): "aaa"
11 "aaa".to_i #Nie wyrzuca wyjątku
12 => 0
```

Notatki











## Przykłady

```
1 def dwarazy
2   yield
3   yield
4 end
5 dwarazy { puts "Witaj" } #można również wywołać tak: dwarazy() { puts "Witaj" }
6 #Na wyjściu:
7   Witaj
8   Witaj
9
10 def dwarazy
11   yield 1
12   yield 2
13 end
14 dwarazy {|i| puts "#{i}. raz: Witaj"}
15 #Na wyjściu:
16   1. raz: Witaj
17   2. raz: Witaj
18
19 def dwarazy(&blok)
20   blok.call 1 # zmienna 'blo
21   blok.call 2
22 end
23 dwarazy {|i| puts "#{i}. raz: Witaj"}
24 #Na wyjściu
25   1. raz: Witaj
26   2. raz: Witaj
27 =begin
28 Można również wywołać tak:
29   dwarazy() {|i| puts "#{i}. raz: Witaj"}
30 ale nie tak:
31   dwarazy( {|i| puts "#{i}. raz: Witaj" } )
32 =end
```

## Notatki

---

---

---

---

---

---

---

---

---

---



## Argumenty bloków

```
1 i = "foo"
2 j = "foo"
3 puts "Przed blokiem: i = #{i}"
4 puts "Przed blokiem: j = #{j}"
5 3.times do |i|
6   puts "i = #{i}"
7   j = i
8   k = 2
9 end
10 puts "Poza blokiem: i = #{i}"
11 puts "Poza blokiem: j = #{j}"
12 puts "Poza blokiem: k = #{k}"
```

## Notatki

---

---

---

---

---

---

---

---

---

---





## Przekształcanie metody w obiekt 'Proc'

## Wybieranie liczb nieparzystych

```

1 #####
2 # Dwuetapowo
3 #####
4 odd = :odd?.to_proc # Zmienna 'odd' zawiera obiekt klasy 'Proc', reprezentujący metodę o nazwie takiej
   jak podany symbol
5 p odd #Wypisze: <Proc:0x...(&:odd?)
6 p [1, 2, 3].select(&odd) #Wypisze: [1, 3]
7 #####
8 # zmienna = :nazwa_metody.to_proc}
9 # obiekt.iterator (&zmienna)
10 #
11 # obiekt.iterator (&nazwa_metody)
12 #####
13 # Jednoetapowo
14 #####
15 p [1, 2, 3].select(&:odd?) #Wypisze: [1, 3]
16 #####
17 # obiekt.iterator (&nazwa_metody) ↔ obiekt.iterator {|arg| arg.nazwa_metody}
18 # operator '&' powoduje przekazanie nazwy metody do iteratora
19 #####

```

Bez użycia bloku — z bezpośrednim użyciem obiektu 'Proc'

```

1 p [1, 2, 3].select{|e| e.odd?} #Wypisze: [1, 3]

```

Z użyciem bloku

Notatki



## Domknięcia

```

1 =begin
2   funkcja wykorzystuje trzy zmienne:
3   'y' - dostępną dla użytkownika
4   'x' - dostępną tylko wewnątrz funkcji 'mnozenie_przez()'
5   'z' - dostępną tylko wewnątrz funkcji 'mnozenie_przez()'
6 =end
7 #####
8 def mnozenie_przez(x)
9   z=0
10  lambda{|y| z+= 1 ; printf "z=%d, %d*%d=",z,x,y; return x*y}
11 end
12 #####
13 iloczyn_5_przez = mnozenie_przez(5) # Parametrowi 'x' jest przypisywana wartość 5
14 iloczyn_6_przez = mnozenie_przez(6) # Parametrowi 'x' jest przypisywana wartość 6
15 #####
16 puts iloczyn_5_przez #Wypisze: #<Proc:0x0000000c0de3801lambda.rb:8 (lambda)>
17 x=10
18 y=20
19 z=30
20 puts iloczyn_5_przez.call(10) # Wypisze: z=1, 5*10=50
21 puts iloczyn_5_przez.call(11) # Wypisze: z=2, 5*11=55
22 puts iloczyn_6_przez.call(12) # Wypisze: z=1, 6*12=72
23 #####
24 puts x #Wypisze: 10
25 puts y #Wypisze: 20
26 puts z #Wypisze: 20

```

Notatki



<sup>2</sup>Fragment kodu, w którym zmienne lokalne zachowują swoją tożsamość

## Definiowanie klas i tworzenie obiektów

Notatki

---



---



---



---



---



---



---



---



---



---

```

1 class Zwierze
2 end
3
4 puts Zwierze.class #Wypisze: Class
5 zwierze = Zwierze.new
6 p zwierze #Wypisze: #<Zwierze:0x00000009c17d8>

```



## Metody

Notatki

---



---



---



---



---



---



---



---



---



---

```

1 class Zwierze
2   @obiekt_self = self
3   #####
4   ##### Metoda instancyjna #####
5   #####
6   def wypisz
7     printf "%s %s\n", self, self.class
8   end
9   #####
10  ##### Metoda klasowa #####
11  #####
12  def Zwierze.wypisz
13    #def self.wypisz
14      printf "%s %s %s %s\n", self, self.class, @obiekt_self, @obiekt_self.class
15    end
16  end
17  #####
18  z=Zwierze.new
19  z.wypisz #Wypisze: #<Zwierze:0x00000002142d58> Zwierze
20  # To co powyzej jest równoważne z.send('wypisz')
21  Zwierze.wypisz #Wypisze: Zwierze Class Zwierze Class
22  # To co powyzej jest równoważne Zwierze.send('wypisz')

```



## Metoda initialize()

Notatki

---

---

---

---

---

---

---

---

---

---

```
1 class Zwierze
2   def initialize(nazwa)
3     puts nazwa
4   end
5 end
6
7 zwierze = Zwierze.new('Pies') #Wypisze: Pies
```

## Dostęp do metody klasowej z wnętrza metody instancyjnej

Notatki

---

---

---

---

---

---

---

---

---

---

```
1 class Zwierze
2   def metodaInstancyjna
3     puts "Jestem metodą instancyjną"
4     self.class.metodaKlasowa
5   end
6
7   def self.metodaKlasowa
8     puts "Jestem metodą klasową"
9   end
10 end
11 #####
12 z = Zwierze.new
13 z.metodaKlasowa #Błąd wykonania: undefined method 'metodaKlasowa' ...
14 #####
15 z.metodaInstancyjna # Wypisze:
16                   # Jestem metodą instancyjną
17                   # Jestem metodą klasową
```



## Dziedziczenie

```

1 class Zwierze #Klasą bazową jest Object
2   def initialize
3     puts 'Konstruktor Zwierze'
4   end
5 end
6 #####
7 p Zwierze.superclass #Wypisze: Object
8 p Zwierze.superclass.superclass #Wypisze:
9   BasicObject
10 p Zwierze.superclass.superclass.superclass #Wypisze:
11   nil

```

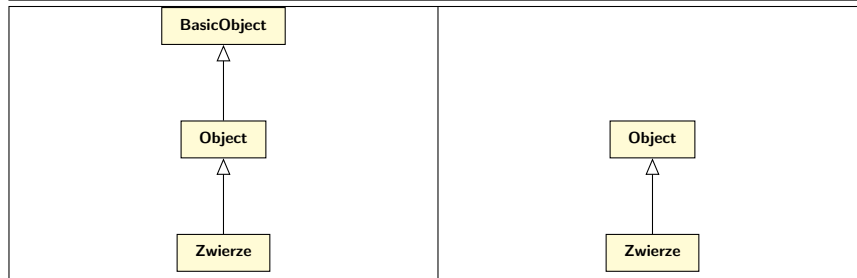
```

10 class Pies < Zwierze
11   def initialize
12     super
13     puts "Konstruktor Pies"
14   end
15 end
16 #####
17 Pies.new #Wypisze:
18   #Konstruktor Zwierze
19   #Konstruktor Pies

```

Hierarchia dziedziczenia w Ruby  $\geq 1.9$

Hierarchia dziedziczenia w Ruby  $\leq 1.8$



Notatki



## Atrybuty

Instancyjne

```

1 class Zwierze
2   @ilosc = 1
3
4   def wypisz_ilosc
5     p @ilosc #Wbrew pozorom, wartością atrybutu '
6       @ilosc' nie jest 1
7   end
8
9   def self.wypisz_ilosc
10    p @ilosc
11  end
12
13  # attr_writer :gatunek
14  def gatunek=(g)
15    @gatunek = g
16  end
17
18  # attr_reader :gatunek
19  def gatunek
20    @gatunek
21  end
22
23  attr_accessor :wiek # attr_accessor ⇔ attr_reader
24    U attr_writer
25 end

```

```

24 z = Zwierze.new
25 z.wypisz_ilosc #Wypisze: nil
26 Zwierze.wypisz_ilosc #Wypisze: 1
27 puts z.ilosc # ... undefined method 'ilosc' for #<
28   Zwierze:0x000000017923d0> (NoMethodError)
29 puts z.@ilosc # ... syntax error, unexpected tIVAR,
30   expecting '('
31 puts @z.ilosc # ... undefined method 'ilosc' for nil:
32   NilClass (NoMethodError)
33 z.gatunek = "Ssak"
34 puts z.gatunek # Wypisze: "Ssak"
35 z.wiek = 10
36 puts z.wiek # Wypisze: 10

```

Notatki



## Atrybuty

## Klasowe

```

1 class Zwierze
2   @@ilosc = 0
3
4   def initialize
5     @@ilosc += 1
6   end
7
8   def ilosc # metoda instancyjna
9     @@ilosc
10  end
11
12  def self.ilosc # metoda klasowa
13    @@ilosc
14  end
15 end
16 #####
17 puts Zwierze.ilosc #Wypisze: 0
18 #####
19 class Pies < Zwierze
20   @@ilosc = 10
21 end
22 #####
23 puts Pies.ilosc #Wypisze: 10
24 puts Zwierze.ilosc #Wypisze: 10
25 pies = Pies.new
26 puts pies.ilosc #Wypisze: 11
27 puts Pies.ilosc #Wypisze: 11
28 puts Zwierze.ilosc #Wypisze: 11

```

## Notatki



## Atrybuty klasowe, a atrybuty instancyjne klasy

## Atrybut klasowy

```

1 class Zwierze
2   # '@@ilosc' jest atrybutem klasowym dzielonym przez klasę '
3   # Zwierze' i każdą z jej podklas
4   # Kiedy stworzysz instancję klasy 'Zwierze' lub jej podklasy,
5   # tj. 'Pies',
6   # to wartość zmiennej 'ilosc' zwiększa się
7   @@ilosc = 0
8
9   def initialize
10    self.class.ilosc += 1 #Wywołanie metod 'self.ilosc=' oraz
11    'self.ilosc'
12  end
13
14  def self.ilosc
15    @@ilosc
16  end
17
18  def self.ilosc=(wartosc)
19    @@ilosc = wartosc
20  end
21 end
22
23 class Pies < Zwierze
24 end
25
26 8.times { Zwierze.new }
27 4.times { Pies.new }
28
29 p Zwierze.ilosc #Wypisze: 12
30 p Pies.ilosc   #Wypisze: 12

```

## Atrybut instancyjny klasy

```

1 class Zwierze
2   # '@ilosc' jest atrybutem instancyjnym (wyłącznie) klasy '
3   # Zwierze'
4   # Wartość zmiennej 'ilosc' ulega zwiększeniu, tylko wtedy
5   # gdy stworzysz instancję klasy 'Zwierze' (nie podklasy,
6   # tj. Pies)
7   @ilosc = 0
8
9   def initialize
10    self.class.ilosc += 1 #Wywołanie metod 'self.ilosc=' oraz
11    'self.ilosc'
12  end
13
14  def self.ilosc
15    @ilosc
16  end
17
18  def self.ilosc=(wartosc)
19    @ilosc = wartosc
20  end
21 end
22
23 class Pies < Zwierze
24   @ilosc = 0
25 end
26
27 8.times { Zwierze.new }
28 4.times { Pies.new }
29
30 p Zwierze.ilosc #Wypisze: 8
31 p Pies.ilosc   #Wypisze: 4

```

## Notatki



## Moduły

```

1 module Math
2   PI = 3.14159265358979
3   def Math.sqrt(x)
4     # ..
5   end
6   ...
7 end
8 #####
9 Math.sqrt(2) # => 1.4142135623731
10 Math::sqrt(2) # => 1.4142135623731
11 Math::PI # => 3.14159265358979
12 #####
13 module Brzeczky
14   def dzwon
15     puts "BZZZZ!BZZZZ!BZZZZ!"
16   end
17 end
18 #####
19 class Czasomierz
20   def podaj_czas
21     puts Time.now
22   end
23 end
24 #####
25 class Budzik < Czasomierz
26   include Brzeczky
27 end
28 #####
29 b = Budzik.new
30 b.podaj_czas # Sun Aug 05 17:24:08 +0200 2007
31 b.dzwon # BZZZZ!BZZZZ!BZZZZ!
32 Budzik.dzwon #undefined method 'dzwon' for Budzik:
   Class (NoMethodError)

```

Dr inż. Stanisław Polak

246

## Notatki

```

1 class Budzik < Czasomierz
2   extend Brzeczky
3 end
4 #####
5 b = Budzik.new
6 b.podaj_czas # Sun Aug 05 17:24:08 +0200 2007
7 b.dzwon # undefined method 'dzwon' for #<Budzik:0
   x0000000242c0c8> (NoMethodError)
8 Budzik.dzwon #BZZZZ!BZZZZ!BZZZZ!
9 zegarek = Czasomierz.new
10 zegarek.extend(Brzeczky)
11 zegarek.dzwon # BZZZZ!BZZZZ!BZZZZ!

```



## Metody klas oraz modułów

## Sterowanie kolejnością wykonywania

```

1 module Brzeczky
2   def podaj_czas
3     puts "Brzeczky: " + Time.now.to_s
4     super
5   end
6 end
7 #####
8 class Czasomierz
9   def podaj_czas
10    puts "Czasomierz: " + Time.now.to_s
11  end
12 end
13 #####
14
15 class Budzik < Czasomierz
16   include Brzeczky
17   def podaj_czas
18     puts "Budzik: " + Time.now.to_s
19     super
20   end
21 end
22
23 Budzik.new.podaj_czas

```

```

1 module Brzeczky
2   def podaj_czas
3     puts "Brzeczky: " + Time.now.to_s
4     super
5   end
6 end
7 #####
8 class Czasomierz
9   def podaj_czas
10    puts "Czasomierz: " + Time.now.to_s
11  end
12 end
13 #####
14
15 class Budzik < Czasomierz
16   prepend Brzeczky #Dostępne w Ruby >= 2.0
17   def podaj_czas
18     puts "Budzik: " + Time.now.to_s
19     super
20   end
21 end
22
23 Budzik.new.podaj_czas

```

## Notatki

Dr inż. Stanisław Polak

247







## Kontrola dostępu

```

1 class Test
2   public # każda metoda (poza initialize(), która jest prywatna) jest domyślnie publiczna
3   def pub_met1
4   end
5
6   private # metoda prywatna
7   def priv_met1
8   end
9
10  protected # wymienione niżej metody są chronione
11  def prot_met1
12  end
13
14  def prot_met2
15  end
16 end

```

```

1 class Test
2   def pub_met1
3   end
4
5   def priv_met1
6   end
7
8   def prot_met1
9   end
10
11  public :pub_met1
12  private :priv_met1
13  protected :prot_met1
14 end

```

Notatki



## Kontrola dostępu

## Przykład użycia

```

1 class Foo
2   def a
3   end
4
5   # Wywołanie 'a()' z jawnym 'self' jako odbiorcą
6   def b
7     self.a
8   end
9
10  # Wywołanie metody 'a()' z niejawnym 'self' jako odbiorcą
11  def c
12    a
13  end
14
15  publicprotectedprivate :a
16 end
17 #####
18 class Foo1 < Foo
19 end
20 #####
21 def safe_send(receiver, method, message)
22   # Nie można używać 'send()', ponieważ omija zasady widoczności
23   eval "receiver.#{method}"
24 rescue => e
25   puts "#{message}: #{e}"
26 else
27   puts "#{message}: wywołanie metody powiodło się"
28 end
29 #####
30 foo = Foo.new
31 foo1 = Foo1.new
32 safe_send(foo, :a, "jawnny odbiorca ")
33 safe_send(foo, :b, "jawnny odbiorca 'self' ")
34 safe_send(foo, :c, "niejawnny odbiorca 'self' ")
35 safe_send(foo1, :a, "jawnny odbiorca ")
36 safe_send(foo1, :b, "jawnny odbiorca 'self' ")
37 safe_send(foo1, :c, "niejawnny odbiorca 'self' ")

```

skrypt.rb

```

1 $ ruby skrypt.rb
2 jawnny odbiorca      : wywołanie metody powiodło się
3 jawnny odbiorca 'self' : wywołanie metody powiodło się
4 niejawnny odbiorca 'self': wywołanie metody powiodło się
5 jawnny odbiorca      : wywołanie metody powiodło się
6 jawnny odbiorca 'self' : wywołanie metody powiodło się
7 niejawnny odbiorca 'self': wywołanie metody powiodło się

```

```

1 $ ruby skrypt.rb
2 jawnny odbiorca      : protected method 'a' called for #<Foo:0
3   x000000028a0960>
4 jawnny odbiorca 'self' : wywołanie metody powiodło się
5 niejawnny odbiorca 'self': wywołanie metody powiodło się
6 jawnny odbiorca      : protected method 'a' called for #<Foo1:0
7   x000000028a0938>
8 jawnny odbiorca 'self' : wywołanie metody powiodło się
9 niejawnny odbiorca 'self': wywołanie metody powiodło się

```

```

1 $ ruby skrypt.rb
2 jawnny odbiorca      : private method 'a' called for #<Foo:0
3   x000000021b4840>
4 jawnny odbiorca 'self' : private method 'a' called for #<Foo:0
5   x000000021b4840>
6 niejawnny odbiorca 'self': wywołanie metody powiodło się
7 jawnny odbiorca      : private method 'a' called for #<Foo1:0
8   x000000021b4818>
9 jawnny odbiorca 'self' : wywołanie metody powiodło się
10 niejawnny odbiorca 'self': wywołanie metody powiodło się

```

Notatki





## Blok jako funkcja wyższego rzędu

Notatki

```
1 open("/etc/passwd") { |plik| puts plik.size }
```

Funkcja, która zawiera „funkcję” na liście argumentów

```
1 def dodawacz(a)
2   proc { |b| a + b }
3 end
4 dodawacz5 = dodawacz(5)
5 puts dodawacz5.call(3) #Wypisze: 8
```

Funkcja, która zwraca „funkcję”



## Częściowe zastosowanie funkcji oraz rozwijanie funkcji

Notatki

```
1 suma = proc { |x, y, z| (x||0) + (y||0) + (z||0) }
2 p suma.call(1, 2, 3) # 6
3 p suma[1, 2, 3] # 6
4 #####
5 # Częściowe zastosowanie funkcji (ang. partial function
6   application)
7 suma_partial = proc { |x, y| proc { |z| x + y + z } }
8 p suma_partial.call(1, 2).call(3) # 6
9 p suma_partial[1, 2][3] # 6
10 #####
11 # Rozwijanie "funkcji" (ang. currying)
12 suma_curry = proc { |x| proc { |y| proc { |z| x + y + z } } }
13 p suma_curry.call(1).call(2).call(3) # 6
14 p suma_curry[1][2][3] # 6
15 #####
16 f = proc { |x, y, z| [x, y, z] }
17 p f[4, 5, 6] # [4, 5, 6]
18 p f[4][5, 6] # nil
19 p f[4, 5][6] # nil
20 p f[4][5][6] # undefined method '[]' for nil:NilClass (
21   NoMethodError)
22 g = f.curry
23 p g[4, 5, 6] # [4, 5, 6]
24 p g[4][5, 6] # [4, 5, 6]
25 p g[4, 5][6] # [4, 5, 6]
26 p g[4][5][6] # [4, 5, 6]
27 p g[4] # #<Proc:0x00562b3a477028>
28
29 gg = f.curry(2)
```

```
30 p gg[4, 5, 6] # [4, 5, 6]
31 p gg[4][5, 6] # [4, 5, 6]
32 p gg[4, 5][6] # nil
33 p gg[4][5][6] # nil
34 p gg[4] # #<Proc:0x00555ebc5edbc8>
35
36 h = g[1] # 'x' otrzymuje wartość 1
37 p h[4, 5, 6] # [1, 4, 5]
38 p h[4, 5] # [1, 4, 5]
39 p h[4][5] # [1, 4, 5]
40 p h[4] # #<Proc:0x000000019404c0>
41
42 hh = gg[1] # 'x' otrzymuje wartość 1
43 p hh[4, 5, 6] # [1, 4, 5]
44 p hh[4, 5] # [1, 4, 5]
45 p hh[4][5] # nil
46 p hh[4] # [1, 4, nil]
47
48 i = h[2] # 'y' otrzymuje wartość 2
49 p i[4, 5, 6] # [1, 2, 4]
50 p i[4, 5] # [1, 2, 4]
51 p i[4] # [1, 2, 4]
52 p i # #<Proc:0x005558a3d2a2b38>
53
54 ii = hh[2] # 'y' otrzymuje wartość 2
55 p ii[4, 5, 6] # wrong number of arguments (given 3,
56   expected 1..2) (ArgumentError)
57 p ii[4, 5] # nil
58 p ii[4] # nil
59 p ii # [1, 2, nil]
60
61 j = i[3] # 'z' otrzymuje wartość 3
62 p j # [1, 2, 3]
63
64 jj = ii[3]
65 p jj # nil
```







## Metoda map()

Nie

```
1 wynik = []
2 [1,2,3,4].each do |x|
3   wynik << x*2
4 end
5 wynik #[2, 4, 6, 8]
```

Tak

```
1 wynik=[1,2,3,4].map do |x|
2   x*2
3 end #[2, 4, 6, 8]
```

Notatki

---

---

---

---

---

---

---

---

---

---



## Metoda select()

Nie

```
1 wynik = []
2 [1,2,3,4].each do |x|
3   wynik << x if x > 2
4 end
5 wynik #[3,4]
```

Tak

```
1 wynik=[1,2,3,4].select do |x|
2   x > 2
3 end #[3, 4]
```

Notatki

---

---

---

---

---

---

---

---

---

---



## Metoda detect()

Notatki

Nie

```
1 wynik = nil
2 [1,2,3,4].each do |x|
3   if x > 2
4     wynik = x
5     break
6   end
7 end
8 wynik #3
```

Tak

```
1 wynik=[1,2,3,4].detect do |x|
2   x > 2
3 end #3
```



## Metoda inject()

Notatki

Nie

```
1 suma = 0
2 [1,2,3,4].each do |x|
3   suma += x
4 end
5 suma #10
```

Tak

```
1 suma = [1,2,3,4].inject(0) do |aku, x|
2   aku+x
3 end #10
4 #####
5 suma = [1,2,3,4].inject(0, :+) #10
```



## Metoda zip()

Notatki

Nie

```

1 a = [1,2,3,4]
2 b = [5,6,7,8]
3 wynik=[]
4 0.upto(a.length-1).each do |i|
5   wynik << [a[i], b[i]]
6 end
7 wynik #[[1, 5], [2, 6], [3, 7], [4, 8]]

```

Tak

```

1 a = [1,2,3,4]
2 b = [5,6,7,8]
3 wynik = a.zip(b) #[[1, 5], [2, 6], [3, 7], [4, 8]]

```



## Wybrane metody do tworzenia iteratorów

Notatki

```

1 [1,2,3].permutation(2).to_a
2 => [[1, 2], [1, 3], [2, 1], [2, 3], [3, 1], [3, 2]]
3 [1,2,3].permutation.to_a
4 => [[1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2], [3, 2, 1]]
5 #####
6 [1,2,3].combination(2).to_a
7 => [[1, 2], [1, 3], [2, 3]]
8 #####
9 [0,1,2,3].group_by {|x| x%2}
10 => {0=>[0, 2], 1=>[1, 3]}
11 [0,0,0,1,1,2].group_by {|x| x}.map {|k,v| v}
12 => [[0, 0, 0], [1, 1], [2]]

```



## Przykład użycia

Notatki

```
1 (1..Float::INFINITY).lazy.select { |x| (x**2 % 5).zero? }.take(10).inject(:+)
2 => 275
```



Dr inż. Stanisław Polak

266

Interfejs Rack

## Użycie „Rack”

Przykład 1

Notatki

```
1 #!/usr/bin/env ruby
2 require "rack"
3 require "rackup"
4
5 class HelloWorld
6   def call(env)
7     sleep(10) #uśpij program na 10 sekund
8     return [200, {}, ["Hello world!"]]
9   end
10 end
11
12 Rackup::Server.start app: HelloWorld.new, Port: 9000
```

helloworld.rb

```
1 $ chmod 755 ./helloworld.rb
2 $ ./helloworld.rb
3 [2014-04-14 10:43:16] INFO WEBrick 1.3.1
4 [2014-04-14 10:43:16] INFO ruby 2.1.1 (2014-02-24)
   [x86_64-linux]
5 [2014-04-14 10:43:16] INFO WEBrick::HTTPServer#
   start: pid=9026 port=9000
6 localhost - - [14/Apr/2014:10:44:04 CEST] "GET /
   HTTP/1.1" 200 12
7 => /
```

```
1 $ curl localhost:9000
2 Hello world!
```



Dr inż. Stanisław Polak

268

Materiały dla studentów wydziału I PŁ AGH w Krakowie

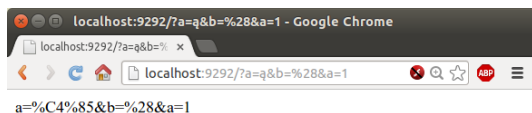


## Obsługa formularzy HTML

```

1 <form action="http://localhost:9292/">
2 <input type="TEXT" name="a" value="a" />
3 <input type="TEXT" name="b" value="(" />
4 <input type="TEXT" name="a" value="1" />
5 <input type="SUBMIT">
6 </form>

```



```

1 class HelloWorld
2   def call(env)
3     odpowiedź = ""
4     #Odczytywanie danych otrzymanych od przeglądarki
5     odpowiedź << env['QUERY_STRING'] #Dane
6       otrzymane (metodą GET) od przeglądarki
7     odpowiedź << env['rack.input'].read #Dane
8       otrzymane (metodą POST) od przeglądarki
9
10    #Wysyłanie odpowiedzi do przeglądarki
11    return [
12      200,
13      {'content-type' => 'text/html'},
14      [odpowiedź]
15    ]
16  end
17 end

```

helloworld.rb

Notatki

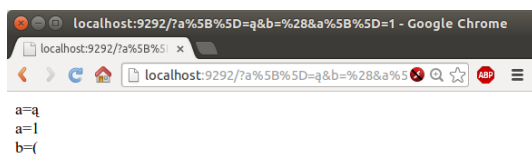
## Obsługa formularzy HTML

Obiekt Rack::Request

```

1 <form action="http://localhost:9292/">
2 <input type="TEXT" name="a[]" value="a" />
3 <input type="TEXT" name="b" value="(" />
4 <input type="TEXT" name="a[]" value="1" />
5 <input type="SUBMIT">
6 </form>

```



```

1 class HelloWorld
2   def call(env)
3     req = Rack::Request.new(env)
4     odpowiedź = [
5       "a="+req.params['a'][0]+"<br>",
6       "a="+req.params['a'][1]+"<br>",
7       "b="+req.params['b'] # zamiast "req.params['b']"
8       można użyć "req['b']"
9     ]
10    return [
11      200,
12      {'content-type' => 'text/html; charset=
13        utf-8'},
14      odpowiedź
15    ]
16  end
17 end

```

helloworld.rb

Notatki





## Aplikacja pośrednicząca

Przekazywanie parametrów

```

1 class ToUpper
2   def initialize(app, param, &blok)
3     puts app
4     puts app.class
5     puts '-' * 10
6     puts param
7     puts param.class
8     puts '-' * 10
9     blok.call('Napis')
10  end
11  ...
12  end
13
14 class Hello
15   ...
16   ...
17 end
18
19 # use(middleware, *args, &block)
20 use ToUpper, 'Argument' { |arg| puts arg }
21 run Hello

```

config.ru

Na wyjściu

Hello

Class

-----

Argument

String

-----

Napis

Notatki

---



---



---



---



---



---



---



---



---



---



## Ogólnodostępne aplikacje pośredniczące

Przykład użycia

```

1 require './helloworld'
2 require 'rack/contrib'
3
4 use Rack::Static, :urls => ["/css", "/images"], :root => "
   public"
5 use Rack::ContentType, "text/plain"
6 use Rack::BounceFavicon
7 use Rack::Reloader, 0
8 run HelloWorld.new

```

config.ru

```

1 <link rel="stylesheet" href="/css/style.
   css">

```

Dokument HTML

```

1 h1 {color: red}

```

public/css/style.css

Notatki

---



---



---



---



---



---



---



---



---



---









## Układ

```

1 <html>
2 <head>
3 <title>Aplikacja Sinatra</title>
4 </head>
5 <body>
6 <h1>Aplikacja Sinatra</h1>
7 <!-- początek wstawki szablonu -->
8 <%= yield %>
9 <!-- koniec wstawki szablonu -->
10 </body>
11 </html>

```

views/layout.erb

```

1 <%= 3.downto(0) do |i| %>
2 <%= i %>,
3 <%= end %>
4 Start

```

views/witaj.erb

```

1 $ curl http://localhost:4567/witaj
2 <html>
3 <head>
4 <title>Aplikacja Sinatra</title>
5 </head>
6 <body>
7 <h1>Aplikacja Sinatra</h1>
8 <!-- początek wstawki szablonu -->
9 3,
10 2,
11 1,
12 0,
13 Start
14
15 <!-- koniec wstawki szablonu -->
16 </body>
17 </html>

```

Wykonywanie żądania

Notatki



## Dostęp do zmiennych z poziomu szablonu

```

1 require 'sinatra'
2
3 get '/witaj' do
4   @tekst1 = "Witaj" #Zmienna instancyjna jest dostępna z poziomu szablonu
5   tekst2 = "Świecie" #Zwykła zmienna nie jest dostępna z poziomu szablonu
6   erb(:witaj, :locals => {:tekst3 => tekst2}) #Definiowanie, dla szablonu 'witaj', zmiennej lokalnej '
   tekst3'
7 end

```

skrypt.rb

```

1 <%= @tekst1 %> <%= tekst3 %>

```

views/witaj.erb

```

1 $ curl http://localhost:4567/witaj
2 Witaj Świecie

```

Wykonywanie żądania

Notatki



## Przykład

Przetwarzanie zawartości formularza

Notatki

```

1 require 'sinatra'
2
3 get '/formularz' do
4   erb :formularz
5 end
6
7 post '/formularz' do
8   "Pole 'nazwa' zawiera napis '#{params[:nazwa]}'"
9 end

```

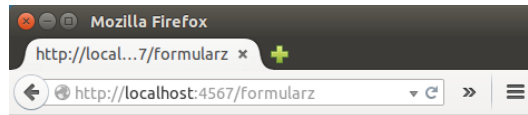
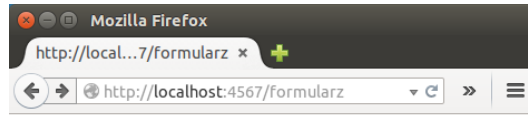
skrypt.rb

```

1 <form action="/formularz" method="post">
2   <input type="text" name="nazwa">
3   <input type="submit">
4 </form>

```

views/formularz.erb



Pole 'nazwa' zawiera napis 'Stanisław Polak'



## Używanie aplikacji pośredniczącej

Notatki

```

1 class ToUpper
2   def initialize(app)
3     @app = app
4   end
5
6   def call(env)
7     status, headers, body = @app.call(env)
8     upcased_body = body.map { |chunk| chunk.upcase }
9     [status, headers, upcased_body]
10  end
11 end

```

to\_upper.rb

```

1 $ curl http://localhost:4567/witaj
2 WITAJ ŚWIECIE

```

Wykonywanie żądania

```

1 require 'sinatra'
2 load 'to_upper.rb'
3
4 use ToUpper
5
6 get '/witaj' do
7   "Witaj Świecie\n"
8 end

```

skrypt.rb





## Źródła I

- ▶ Jamis Buck. *Method visibility in Ruby*. URL: <http://weblog.jamisbuck.org/2007/2/23/method-visibility-in-ruby>.
- ▶ Radosław Bułat. *Blog*. URL: <http://radarek.jogger.pl/>.
- ▶ *Getting Started*. URL: <https://reactjs.org/docs/getting-started.html>.
- ▶ Paweł Grzesiak. *Ajax w kilka minut*. URL: [http://internetmaker.pl/arttykul/723,1,ajax\\_w\\_kilka\\_minut.html](http://internetmaker.pl/arttykul/723,1,ajax_w_kilka_minut.html).
- ▶ Patrick Hunlock. *Functional Javascript*. URL: [http://www.hunlock.com/blogs/Functional\\_Javascript](http://www.hunlock.com/blogs/Functional_Javascript).
- ▶ MongoDB Inc. *The MongoDB Manual*. URL: <http://docs.mongodb.org/manual/>.
- ▶ Joyent. *Node.js Manual & Documentation*. URL: <http://nodejs.org/api/>.
- ▶ Agnieszka Matysek. *Dopasowywanie wzorców - nowa funkcjonalność w Ruby 2.7*. URL: <https://womannonrails.com/pl/ruby-pattern-matching>.
- ▶ Microsoft. *TypeScript handbook*. URL: <http://www.typescriptlang.org/Handbook>.
- ▶ Blake Mizerany. *Sinatra - documentation*. URL: <http://www.sinatrarb.com/documentation.html>.
- ▶ Michael Morin. *Using Rack*. URL: <http://ruby.about.com/od/rack/a/Using-Rack.htm>.

Notatki

---

---

---

---

---

---

---

---

---

---

---

---



## Źródła II

- ▶ Mozilla. *AJAX*. URL: <https://developer.mozilla.org/pl/AJAX>.
- ▶ mozilla.org. *JavaScript Guide*. URL: <https://developer.mozilla.org/en/JavaScript/Guide>.
- ▶ Aleksander Pohl. *Ruby intro*. URL: <http://www.apohllo.pl/dydaktyka/ruby/intro/>.
- ▶ Axel Rauschmayer. *Iterables and iterators in ECMAScript 6*. URL: <http://www.2ality.com/2015/02/es6-iteration.html>.
- ▶ Arnau Sanches. *Functional Programming with Ruby*. URL: <http://www.slideshare.net/tokland/functional-programming-with-ruby-9975242>.
- ▶ Marek Stępień. *marcoos.techblog*. URL: <http://blog.marcoos.com/>.
- ▶ Basarat Ali Syed. *TypeScript Deep Dive*. URL: <http://basarat.gitbooks.io/typescript/>.
- ▶ Techotopia.com. *Ruby Essentials*. URL: [http://www.techotopia.com/index.php/Ruby\\_Essentials](http://www.techotopia.com/index.php/Ruby_Essentials).
- ▶ Kazuki Tsujimoto. *Pattern matching — New feature in Ruby 2.7*. URL: [https://speakerdeck.com/k\\_tsj/pattern-matching-new-feature-in-ruby-2-dot-7](https://speakerdeck.com/k_tsj/pattern-matching-new-feature-in-ruby-2-dot-7).
- ▶ wikibooks.org. *JavaScript*. URL: <http://en.wikibooks.org/wiki/JavaScript>.
- ▶ *Wikipedia*. URL: <http://pl.wikipedia.org/>.
- ▶ *Wikipedia. Kurs PHP*. URL: <http://pl.wikibooks.org/wiki/PHP>.

Notatki

---

---

---

---

---

---

---

---

---

---

---

---





## Źródła III

- ▶ [Wikipedia. \*Kurs Ruby\*. URL: <http://pl.wikibooks.org/wiki/Ruby>.](http://pl.wikibooks.org/wiki/Ruby)
- ▶ [Noppakun Wongsrinoppakun. \*An Array of Possibilities: A Guide to Ruby Pattern Matching\*. URL: <https://www.toptal.com/ruby/ruby-pattern-matching-tutorial>.](https://www.toptal.com/ruby/ruby-pattern-matching-tutorial)

Notatki

---

---

---

---

---

---

---

---

---

---

---

---



Notatki

---

---

---

---

---

---

---

---

---

---

---

---