

Joó Ádám:

CoffeeScript: JavaScript++

A JavaScript napjaink legizgalmasabb
programozási nyelve.

A JavaScript napjaink legizgalmasabb programozási nyelve.

Ezen izgalmak jó része a tervezésére és implementálására szánt körülbelül két hét eredménye.

A JavaScript napjaink legizgalmasabb programozási nyelve.

Ezen izgalmak jó része a tervezésére és implementálására szánt körülbelül két hét eredménye.

De ezektől most tekintsünk el.
Mármint tényleg.

Unearthing the excellence in JavaScript



JavaScript
The Good Parts

Douglas Crockford:
JavaScript:
The Good Parts

CoffeeScript: JavaScript += 1

0.1.0:

2009. december 13.

0.7.2:

2010. július 12.

0.1.0:

2009. december 13.

0.7.2:

2010. július 12.

Jeremy Ashkenas, et al.

Tartalmaz:

- JavaScript Good Parts
- Szintakszis, amibe beleszeretsz
- Modern nyelvi konstrukciók

Tartalmaz:

- JavaScript Good Parts
- Szintakszis, amibe beleszeretsz
- Modern nyelvi konstrukciók

3 Ruby, Python

Fordítás után

- Doug Crockford a barátja
- JSLint a barátja
- JS könyvtárak a baráta

Sorról-sorra
megfeleltetve
(majdnem)



Fordítás után

- Doug Crockford a barátja
- JSLint a barátja
- JS könyvtárak a barátai

Értékkadás, sorvég

```
text: 'Hello Bp.js'
```

```
math = 20 +  
       20 +  
       2
```

Értékkadás, sorvég

```
text: 'Hello Bp.js';
```

```
math = 20 +  
       20 +  
       2;
```

Értékkadás, sorvég

```
text: 'Hello Bp.js';
```

```
math = 20 +  
       20 +  
       2;
```

```
text = 'Hello Bp.js';
```

```
math = 20 + 20 + 2;
```

Értékadás, sorvég

```
text: 'Hello Bp.js';
```

```
math = 20 +  
       20 +  
       2;
```

```
text = 'Hello Bp.js';
```

```
math = 20 + 20 + 2;
```



objektumliterálban is

Függvények, blokkok, hatókör

```
polite: (name) ->  
  say: 'Hello '
```

```
  alert say + name
```

Függvények, blokkok, hatókör

```
polite: (name) -> {  
  var say: 'Hello '  
  
    alert(say + name)  
}
```

Függvények, blokkok, hatókör

```
polite: (name) -> {  
  var say: 'Hello '  
  
  alert(say + name)  
}
```

```
polite = function (name) {  
  var say;  
  
  say = 'Hello '  
  
  alert(say + name);  
}
```

Függvények, blokkok, hatókör

```
polite: (name) -> {  
  var say: 'Hello '  
  
  alert(say + name)  
}
```

```
polite = function (name) {  
  var say;  
  
  say = 'Hello '  
  
  alert(say + name);  
}
```

Az indentálás működik
feltételvizsgálattal és
ciklusokkal is

Tömbök, objektumok

```
inline: [1, 2, 3, 4]
```

```
multiline: [  
    1, 2  
    3, 4  
]
```

```
object: {  
    foo    : 'x'  
    bar    : 'y'  
    class  : 'z'  
}
```

Tömbök, objektumok

```
inline: [1, 2, 3, 4]
```

```
multiline: [  
    1, 2,  
    3, 4,  
]
```

```
object: {  
    foo    : 'x',  
    bar    : 'y',  
    'class': 'z'  
}
```

Tömbök, objektumok

```
inline: [1, 2, 3, 4]
```

```
inline = [1, 2, 3, 4];
```

```
multiline: [  
    1, 2,  
    3, 4,  
]
```

```
multiline = [1, 2, 3, 4];
```

```
object: {  
    foo    : 'x',  
    bar    : 'y',  
    'class': 'z'  
}
```

```
object: {  
    foo:    'x',  
    bar:    'y',  
    'class': 'z'  
}
```

Feltételvizsgálat

```
if x >= 1  
    alert '+'
```

```
unless x > -1  
    alert '-'
```

```
if 1 > x > -1 then alert '0'
```

```
bool: if y then true else false
```

```
good: or evil
```

Feltételvizsgálat

```
if (x >= 1)
  alert '+'
```

```
unless (x > -1)
  alert '-'
```

```
if (1 > x > -1) then alert '0'
```

```
bool: if y then true else false
```

```
good: or evil
```

Feltételvizsgálat

```
if (x >= 1)
    alert '+'
```

```
unless (x > -1)
    alert '-'
```

```
if (1 > x > -1) then alert '0'
```

```
bool: if y then true else false
```

```
good: or evil
```

```
if (x >= 1) {
    alert('+');
}
```

```
if (!(x >= 1)) {
    alert('-');
}
```

```
if ((1 > x) && (x > -1)) {
    alert('0');
}
```

```
bool = y ? true : false;
```

```
good = good || evil
```

If és unless posztfix
is működik

Feltételvizsgálat

```
if (x >= 1)
    alert '+'
```

```
unless (x > -1)
    alert '-'
```

```
if (1 > x > -1) then alert '0'
```

```
bool: if y then true else false
```

```
good: or evil
```

```
if (x >= 1) {
    alert('+');
}
```

```
if (!(x >= 1)) {
    alert('-');
}
```

```
if ((1 > x) && (x > -1)) {
    alert('0');
}
```

```
bool = y ? true : false;
```

```
good = good || evil
```

Elágazás

```
switch drink
  when 'Beer', 'Wine'
    return 'alcohol'
  when 'Coffee' then return 'caffeine'
  else return 'depression'
```

Elágazás

```
switch (drink) {  
  when 'Beer', 'Wine'  
    return 'alcohol' break;  
  when 'Coffee' then return 'caffeine' break;  
  else return 'depression'  
}
```

Elágazás

```
switch (drink) {  
  when 'Beer', 'Wine'  
    return 'alcohol' break;  
  when 'Coffee' then return 'caffeine' break;  
  else return 'depression'  
}
```

```
if (drink === 'Beer' || drink === 'Wine') {return 'alcohol';}  
else if (drink === 'Coffee') {return 'caffeine';}  
else {return 'depression';}
```

Ciklusok

```
eat()   while hungry  
sleep() until fresh
```

```
loop endless()
```

```
while (hungry) { eat(); }  
while (!fresh) { sleep(); }
```

```
while (true) { endless(); }
```

Kivételkezelés

```
try
  errorProne()
catch error
  alert error
finally
  cleanUp()
```

```
try {
  errorProne();
} catch (error) {
  alert(error);
} finally {
  cleanUp();
}
```

Operátorok, értékek

==	===
!=	!==
is	===
isnt	!==
not	!
and	&&
or	
yes	true
no	false
on	true
off	false

Tulajdonságok

`this.property` helyett használható `@property`

Tömbelemek és objektumkulcsok

```
answers: [yes, yes, no, yes]  
alert 'disagreement' if no in answers
```

```
foobars = {  
    foos: 81  
    bars: 56  
}
```

```
doSomething() if foos of foobars
```

Egzisztencia

```
if warning?  
  alert warning
```

```
if (typeof warning !== "undefined"  
    &&  
    warning !== null  
    )  
{  
  alert(warning);  
}
```

```
speed ?= 140
```

```
speed = (typeof speed !== "undefined"  
&& speed !== null) ? speed : 140;
```

```
user?.address?.zipcode
```

Tömbszeletek

```
numbers: [0, 1, 2, 3, 4, 5]
```

```
numbers[1..3] // [0, 1, 2, 3]
```

```
numbers[1...3] // [0, 1, 2]
```

```
numbers[1..3]: [3, 2, 1] // [0, 3, 2, 1, 4, 5]
```

Felbontó értékadás

$[x, y]: [y, x]$

Iterátorok

```
items = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
for item in items by 2  
    alert item
```

Sztring és regex interpoláció

author: "Wittgenstein"

quote: "A picture is a fact. -- \$author"

Eszközeink

- Fordító
- Interaktív shell
- Node.js modul
- `<script type="text/coffeescript"></script>`

<http://coffeescript.org>

Na?

Joó Ádám

- ceriak@gmail.com
- <http://jooadam.hu>

Cardo by David Perry

Cantarell by Dave Crossland

Droid Sans Mono by Steve Matteson

Reenie Beanie by James Grieshaber

Tangerine by Toshi Omagari