



Osnove programiranja

Dragi polaznici, ova prezentacija nije primarno namijenjena za učenje već služi kao **pomoćni materijal**. Kao takva, ne može zamijeniti predavanja, literaturu kao niti vašu (pro)aktivnost na nastavi i konzultiranje s predavačem u slučaju potencijalnih pitanja i/ili nejasnoća.

Osnove programiranja

5 sati predavanja, 15 sati vježbi

Cjeline

Leksička struktura jezika

Deklaracije

Vježba 1

Tipovi podataka

Rad sa tekstom

Operatori

Izjave

Vježba 2

Nizovi

Funkcije

Objekti

Vježba 3

Standardni ugrađeni objekti

Uvod u napredne koncepte

Osnove programiranja

Leksička struktura jezika

Leksička struktura

Leksička struktura je skup elementarnih pravila koja određuju kako pišete programe na tom jeziku.

Određuje:

- ulogu praznina u jeziku
- kako izgledaju nazivi varijabli
- kako pišemo komentare
- kako se jedna programska izjava odvaja od sljedeće

Elementi sintakse

Vrijednosti - literali i varijable

Operatori - aritmetički, operator dodjele vrijednosti ...

Izrazi - kombinacija vrijednosti i operatora koje se izračunavaju

Ključne riječi - npr. var, return, function, new itd.

Komentari - za objašnjenje koda

Razmaci i indentacija - za lakše čitanje koda

Zagrade

Izjava

Instrukcija koju računalno izvodi.

Naš računalni program je kolekcija izjava.

Izjave se sastoje od:

- vrijednosti
- operatera
- izraza
- ključnih riječi
- komentara

```
var a, b, c;
```

```
a = 5;
```

```
b = 6;
```

```
c = a + b;
```

IZJAVA

PROGRAM

Zagrade

Vitičaste zagrade uvijek se trebaju koristiti, čak i u slučajevima kada nisu obavezni. To čini naš kod dosljednim i lakšim za ažuriranje.

Zagrade određuju **blok** izjavu, koja grupira nula ili više izjava.

```
if (x)
    console.log('asdf');

if (x) {
    console.log('asdf');
}

if (x) {
    console.log('asdf');
    // neka druga izjava
}
```

Indentacija

JavaScript **ignorira indentaciju**, ali radi lakšeg čitanja kôda ju koristimo, i ona bi trebala biti konzistentna.

Postavite pravilo
indentacije u
vašem text editoru.

```
if ('this_is' == /an_example/) {  
  of_beautifier();  
} else {  
  var a === b ? (c % d) : e[f];  
}
```

```
if ('this_is' == /an_example/) {  
    of_beautifier();  
} else {  
    var a === b ? (c % d) : e[f];  
}
```

Točka-zarez

JavaScript upotrebljava točku-zarez za odvajanje izjava, ali ona je neobavezna u slučaju da su izjave u zasebnim linijama.

Mi ćemo uvijek upotrebljavati točku-zarez na kraju naših izjava, radi konzistencije i da izbjegnemo krivu interpretaciju našeg koda.

Leksička struktura jezika - vježba

Pratite upute u Upute.txt

Trajanje vježbe: 25min

Osnove programiranja

Deklaracije

Vrijednosti

Literali su nepromijenjive vrijednosti, poput brojeva i teksta.

Varijable su promjenjive, i koriste se za spremanje podataka.

- koriste ključnu riječ **var** za deklaraciju
- koriste znak jednakosti **=** za dodjeljivanje vrijednosti
- puni izraz za varijablu **x**

var x = 1;

Deklaracija

Prije upotrebe varijable u programu JavaScript, trebali biste je kreirati. Kreacija varijable se službeno zove **deklaracija**. Dodjeljivanje vrijednosti se zove **inicijalizacija**.

Varijable su deklarirane s ključnom riječi **var**.

```
var i;  
var a = 1;
```

Deklaracija

```
// Deklaracija zasebnih varijabi  
var i;  
var sum;
```

```
// U istom retku  
var i, sum;
```

```
// Deklaracija i inicijalizacija  
var i = "tekst";  
var sum = 20;
```

Koja je vrijednost
varijabli i i sum
nakon deklaracije?

Imenovanje

Veliko i malo slovo

JavaScript je jezik razlikuje velika i mala slova. To znači da jezične ključne riječi, varijable, imena funkcija i drugi identifikatori moraju uvijek biti upisani s dosljednim velikim slovima.

mojaVarijabla je različita od **mojavarijabla**

Imenovanje

Identifikatori

Identifikator je jednostavno ime za imenovanje varijabli i funkcija i za pružanje oznaka za određene petlje u JavaScript kodu. Ima par pravila kod imenovanja:

1. Mora početi sa **slovom**, **podvlakom** (`_`) ili **znakom dolara** (`$`)
2. Sljedeći znakovi mogu biti **slova**, **znamenke**, **podvlake** ili `$`

Imenovanje

Rezervirane riječi

JavaScript zadržava određeni broj identifikatora kao ključne riječi samog jezika.

break, delete, case, do, if, switch, var, catch, else, in, this, void, continue, new, null, for, with, default, NaN, Number, String ...

Ove se riječi ne smiju koristiti kao identifikatori!

Komentari

Komentari se zanemaruju prilikom evaluacije koda.

JS podržava dva tipa komentara:

1. sa `//` znakovima - može se koristiti samo u jednom redu
2. između znakova `/*` i `*/` - može se koristiti u više redova

```
// Komentar u jednom retku
```

```
/* Još jedan komentar */
```

```
/* Komentar u više redova  
 * Extra zvjezdice u svakom retku  
 * se koriste u vizualne svrhe,  
 * ali su neobavezna  
 */
```

Komentari

JSDoc

Varijanta Javadoc standarda komentiranja koji služi objašnjenju API-ja funkcije ili koda koji stvaraju (sučjenja prema van koji mogu drugi koristiti).

```
/**
 * Creates an instance of Circle.
 *
 * @constructor
 * @author: Ivan
 * @this {Circle}
 * @param {number} r radius of the circle
 */

function Circle(r) {
    this.radius = r;
    this.circumference = 2 * Math.PI * r;
}
```

Deklaracije - vježba

Pratite upute u Upute.txt

Trajanje vježbe: 20min

Osnove programiranja - Leksička struktura

Vježba

Leksička struktura - vježba

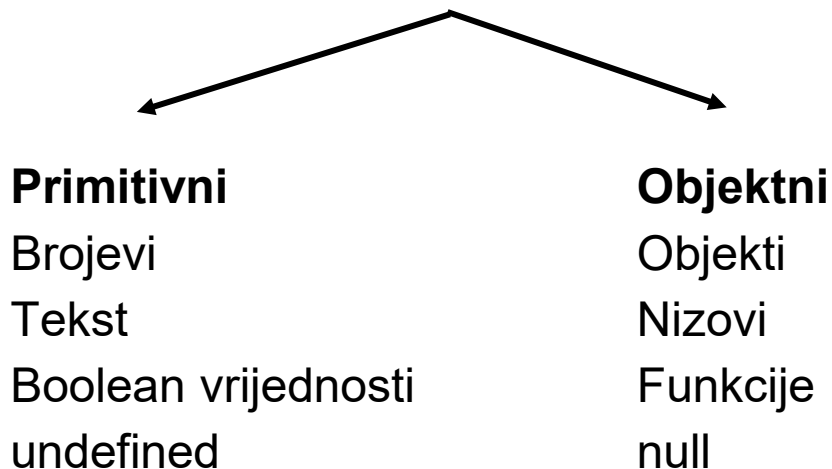
Pratite upute u Upute.txt

Vrijeme trajanja vježbe: 45min

Osnove programiranja

Tipovi podataka

Tipovi podataka u JavaScriptu



Promjenjive i nepromjenjivi tipovi

Također se mogu kategorizirati kao promjenjivi i nepromjenjivi tipovi.

Nepromjenjivi tipovi podatka su oni čije se stanje ne može promijeniti nakon što se podatak stvori.

Primitivni tipovi podataka (i null) su **nepromjenjivi**.

Objektni tipovi podataka su **promjenjivi**.

Brojevi

Integer je cjelobrojna vrijednost (bez decimalnih vrijednosti).

Float je broj sa decimalnom vrijednosti, obično zauzima više memorijskog prostora.

Svi brojevi u JavaScriptu su **floating point** tip broja, i spremaju se u 64-bitnom zapisu!

Tekst

Tekst se u JavaScriptu sprema u **string** tipu podatka.
Prepoznavamo ga po navodnicima.

```
var x = 'Lorem ipsum';  
var y = "Lorem ipsum dolor sit amet";
```

Boolean tip podatka

Predstavlja istinu ili neistinu, uključeno ili isključeno, da ili ne. Postoje samo dvije moguće vrijednosti ove vrste: **true** ili **false**.

Truthy i **falsey** vrijednosti su one koje se konvertiraju u true i false kada izraz očekuje Boolean vrijednost.

```
var x = Boolean(null);    // x = false  
var x = Boolean({});      // x = true
```

null i undefined

null je ključna riječ koja se obično koristi da naznači odsutnost neke vrijednosti.

undefined je vrijednost varijabli koje nisu inicijalizirane i vrijednost koju dobivate kada pokušavate dohvatiti vrijednost objekta ili elementa niza koji ne postoji. Također ćete dobiti undefined za izvršenje funkcije koja ne vraća nikakvu vrijednost.

typeof operator

Možete koristiti operator `typeof` da biste pronašli vrstu podataka.

```
typeof "John"           // "string"
typeof 3.14              // "number"
typeof NaN               // "number"
typeof false             // "boolean"
typeof [1,2,3,4]         // "object"
typeof {name:'John', age:34} // "object"
```


Automatska konverzija podataka

JavaScript obavlja mnoge konverzije tipa podatka automatski ovisno o kontekstu u kojemu se podatak koristi, te operatoru.

```
> '5' - 3
< 2
> '5' + 3
< "53"
> '5' - '4'
< 1
> '5' + + '5'
< "55"
> 'foo' + + 'bar'
< "fooNaN"
> '5' + - '2'
< "5-2"
> '5' + - + - - + - - + + - + - + - - - '2'
< "5-2"
> x = 3
< 3
> '5' + x - x
< 50
> '5' - x + x
< 5
```

Tipovi podataka - vježba

Pratite upute u Upute.txt

Vrijeme trajanja vježbe: 20min

Osnove programiranja

Rad sa tekstom

String

String je niz znakova (0 ili više) okružen navodnicima.

Navodnici mogu biti jednostruki ili dvostruki.

Brojanje pozicije znaka u stringu počinje od **nule!**

```
var myString = "";           // prazan string  
myString = "Lorem ipsum";  
myString = "  Lorem ipsum"; // string sa prazninama na početku
```

Konkatenacija

Operacija spajanja dva ili više stringova.

U JavaScriptu se koristi operator **+**

```
x = "Hello, " + "World";
```

```
x = "Hello, " + 5;
```

```
var x = "Ovo je vrlo dugi string" +  
        "koji je u dva reda " +  
        "jer se inače ne može čitati.";
```

```
var x = "Ovo je vrlo dugi string \  
        "koji je u dva reda \  
        "jer se inače ne može čitati.";
```

Posebni znakovi

Neki znakovi unutar stringa se moraju "izbjeći" kako bi se prikazali.
To radimo sa znakom \ (backslash).

\'	Jednostruki navodnici
\"	Dvostruki navodnici
\\	Backslash
\n	Nova linija
\r	Carriage Return

```
var x = "Ovo je moj \"string\"";
```

Metode

length	Duljina stringa
charAt	Vraća znak na poziciji
trim	Miče praznine na početku i kraju stringa
indexOf	Vraća poziciju stringa unutar stringa
split	Pretvara string u niz - više o tome kasnije!
replace	Vraća novi string sa zamijenjenim izrazom
toUpperCase	Pretvara stringu u sva velika slova

Pazite! Neke metode mijenjaju izvorni string, a neke vraćaju novi string.

Rad sa tekstom - vježba

Pratite upute u Upute.txt

Vrijeme trajanja vježbe: 30min

Osnove programiranja

Operatori

Tipovi operatora

Po tipu operacije

Logički	&& !
Arifmetički	+ - * / % ** ++ --
Operatori dodjeljivanja	= += -= *= /= %= **=
Relacijski	== === != !== > < >= <= ?
Bitovni (bitwise)	& ~ ^ << >> >>>
Operator konkatencije	+
Ostali	delete, typeof, void, instanceof, in

Tipovi operatora

Po broju operandata

1. Unarni
2. Binarni
3. Ternarni - kombinira 3 operacije u jednu

```
var x += 1; // Unarni operator
var x = y + 1; // Binarni operator
var x = y ? y : x + 1; // Ternarni operator
```

Aritmetički operatori

-	oduzimanje	$x = x - 1$	$x -= 1$
+	zbrajanje	$x = x + 1$	$x += 1$
*	množenje	$x = x * 2$	$x *= y$
/	podjela	$x = x / 2$	$x /= 2$
%	modulo (ostatak dijeljenja)	$x = x \% 2$	$x \% = 2$
**	potenciranje (ES6)	$x = x ** 2$	$x ** = 2$
--	dekrement	$x--$ ili $--x$	Različito ponašanje ovisno o kontekstu
++	inkrement	$x++$ ili $++x$	

Relacijski operatori

Testiraju relativni poredak (brožčani ili abecedni) njihova dva operanda:

==	jednaka vrijednost
===	jednaka vrijednost i jednaki tip podatka
!=	Nije jednako
!==	nije jednaka vrijednost ili nije jednaki tip podatka
>	više od
<	manje od
>=	više ili jednako
<=	manje ili jednako

Logički izrazi

&&	i
	ili
!	ne

$$\neg(p \ \&\& \ q) \equiv \neg p \ || \ \neg q$$

$$\neg(p \ || \ q) \equiv \neg p \ \&\& \ \neg q$$

Operatori - vježba

Pratite upute u Upute.txt

Vrijeme trajanja vježbe: 30min

Osnove programiranja

Izjave

Vrste izjava

Uvjeti izvršavaju ili preskaču druge izjave ovisno o vrijednosti određenog izraza, tj. granaju kod na više mogućnosti.

Petlje repetitivno izvršavaju dijelove koda.

Izjave **skokova** (poput break, return, throw) naređuju interpreteru da skoči na neki drugi dio programa.

Uvjeti

If, else if

Ako je izraz truthy, onda se izvrši izjava/e unutar zagrada. Inače se izjava preskače.

Else nam daje alternativu za slučaj da je izraz falsey.

Možemo koristiti i **else if** ako želimo postaviti dodatan uvjet.

```
// If - else
if (izraz) {
    // Izvrši kod u bloku #1
} else {
    // Izvrši kod u
    bloku #2
}
```

Uvjeti

If, else if

```
// Jednostavan if
if (izraz) {
    // Izvrši kod u bloku #1
}
```

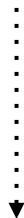
```
// If - else if
if (izraz) {
    // Izvrši kod u bloku
    #1
} else if (izraz2) {
    // Izvrši kod u
    bloku #2
}
```

Uvjeti

switch

Switch izjava je bolja alternativa kompliciranim if izjavama.

```
switch(n) {  
    case 1: // Execute code block #1. break;  
    case 2: // Execute code block #2. break;  
    default: // Execute code block #3. break;  
}
```



Smjer provjere

Petlje

while, do ... while

```
var count = 0;
while (count < 10) {
    console.log(count);
    count++;
}
```

```
var count = 0;
do {
    console.log(count);
    count++;
} while (count < 10);
```

Petlje

for, for ... in

```
for (var count = 0; count < 10; count++) {  
    console.log(count);  
}
```


For ... in petlja se koristi sa objektnim tipovima podataka.

```
for (var i = 0; i < a.length; i++) {  
    console.log(a[i]);  
}
```


Izjave

break, continue

```
for (i = 0; i < 10; i++) {  
    if (i === 3) { continue; }  
    text += "The number is " + i + "<br>";  
}
```



```
for (i = 0; i < 10; i++) {  
    if (i === 3) { break; }  
    text += "The number is " + i + "<br>";  
}
```



continue izjava preskače obradu trenutnog člana petlje i nastavlja sa sljedećim

break izjava prekida petlju i izlazi iz nje

Izjave - vježba

Pratite upute u Upute.txt

Vrijeme trajanja vježbe: 20min

Osnove programiranja - 1.dio

Vježbe

Zadaci

Pratite Upute.txt

Trajanje vježbe: 45min

Osnove programiranja

Nizovi

Niz (array)

Niz je sortirana zbirka vrijednosti.

Svaka pojedina vrijednost u nizu se naziva elementom niza. Može biti bilo kojeg JS tipa podatka, a sam niz je tipa objekt. Niz prepoznavamo po uglatim zagradama ([...]).

```
var name1 = 'John';  
var name2 = 'Jane';  
var name3 = 'Mike';
```

```
var names = ['John', 'Jane', 'Mike'];  
  
console.log(names);  
console.log(typeof(names));
```

Niz

Kreiranje niza

literal

```
var emptyArray = [];  
var weirdArray = [1,,3];  
var names = ['John', 'Jane', 'Mike'];
```

new Array()

```
var emptyArray = new Array();  
var arrayLength5 = new Array(5);  
var names = new Array('John', 'Jane', 'Mike');
```

Elementi niza

Postavljanje i dohvaćanje

Za razliku od klasičnih objekata, elementi u nizu se dohvaćaju brojkom (ne-negativnim intergerom), tj. **indexom** elementa. Indexi elemenata počinju od **nule**.

Nizovi se mogu proširivati i smanjivati po volji.

```
var names = ['John', 'Jane', 'Mike'];  
var name1 = names[0];
```

```
names[3] = 'Holly';
```

Metode nad nizom

push	Dodaje element na kraj niza
pop	Miće zadnji element niza
join	Pretvara sve elemente niza u konkatenirani string
reverse	Obrne redosljed elemenata u nizu
sort	Sortira elemente niza (default: po abecedi)
forEach (ES5)	Iterira kroz niz, pozivajući funkciju koju odredite za svaki element
map (ES5)	Prosljeđuje svaki element niza funkciji i vraća niz koji sadrži vrijednosti vraćene tom funkcijom

Multidimenzionalni nizovi

Za stvaranje dvo-dimenzionalnih matrica, u JavaScriptu jednostavno koristimo niz nizova.

```
var matrix = [[1, 2, 3], ['John', 'Jane', 'Mike']];
```

```
matrix[0][1];  
matrix[2][2];  
matrix[0][4] = 5;  
matrix[1] = [];
```

Koje su vrijednosti ovih izjava?
- Koja je duljina matrix varijable
nakon zadnje izjave?

Iteracije nad nizom

for, forEach

```
var names = ['John', 'Jane',,, 'Mike'];

for (var i = 0; i < names.length; ++i) {
    console.log(names[i]);
}

names.forEach(function(value){
    console.log(value);
});
```

Zadaća: Proučite i for/of i for/in metode iteracija nad nizom, te njihove prednosti/nedostatke.

Nizovi - vježba

Pratite upute u Upute.txt.

Trajanje vježbe: 25min

Osnove programiranja

Funkcije

Funkcija

Funkcija je blok JavaScripta koda koji izvršava neki zadatak.

U drugim jezicima funkcija se naziva procedura ili subrutina.

Funkcije koristimo jer njezin kôd možete **ponovo upotrijebiti**: definirajte ga jednom i koristite ga više puta. Možete koristiti isti kôd **više puta s različitim argumentima** za dobivanje različitih rezultata.

Definicija i pozivanje funkcije

```
function imeFunkcije() {  
    // nesto napravi  
}  
  
// neki jos kod između  
  
imeFunkcije();
```

Definicija i pozivanje funkcije

DEFINICIJA

```
function imeFunkcije() {  
    // nesto napravi  
}
```

POZIVANJE

```
imeFunkcije();
```

“Return” ključna riječ

Svaka funkcija vraća vrijednost undefined, osim ako upotrijebimo ključnu riječ return. Vraća izračunatu vrijednost iz funkcije.

U slučaju izjave return prije kraja funkcije, ostatak funkcije je ignoriran.

```
function imeFunkcije() {  
    var value = 1;  
    return value;  
}  
console.log(imeFunkcije());
```

```
function imeFunkcije() {  
    var value = 1;  
    return value;  
    value += 1;  
}  
console.log(imeFunkcije());
```

Što se ispisuje u konzoli u ova dva slučaja?

Parametri i argumenti

Definicija funkcije može (ali ne mora) sadržavati popis identifikatora, poznatih kao parametri, koji djeluju kao lokalne varijable za tijelo funkcije.

PARAMETRI

```
function udaljenostKoordinata(x1, y1, x2, y2) {  
    var dx = x2 - x1;  
    var dy = y2 - y1;  
    console.log(Math.sqrt(dx*dx + dy*dy));  
}
```

ARGUMENTI

```
udaljenostKoordinata(1,1,3,4);
```


Parametri

JavaScript ne zahtijeva tip podataka u parametrima koje dobiva u funkciji, kao ni točan broj parametara. U funkciji možemo računati na te opcionalne parametre.

```
function myFunc(a, b) {  
    if (!b) {  
        b = 1;  
    }  
    return a + b;  
}
```

```
myFunc(1);
```

```
function myFunc(a, b) {  
    b = b || 1;  
    return a + b;  
}
```

```
myFunc(1,1);
```

```
function myFunc(a, b = 1) {  
    return a + b;  
}
```

```
myFunc(1,1);
```

samo ES6

Funkcija kao vrijednost

Funkcija se u JavaScriptu može koristiti kao bilo koja druga varijabla.

```
var addOne = function(value) {  
    return value + 1;  
};
```

```
var myVar = addOne(1);
```

Koja je vrijednost myVar
varijable u ova dva slučaja?

Funkcije - vježba

1. Riješite zadatke sa slideova sa ispisom u konzoli. Trajanje vježbi: 10min
2. Pratite upute u Upute.txt. Trajanje vježbe: 15min

Osnove programiranja

Objekti

Objekt

Zbirka svojstava od kojih svako ima ime (ključ) i vrijednost. Objekt se prepoznaje po vitičastim zagradama.

```
var dog = {  
    'Golden retriever',  
    bark: function() {  
        return 'Wuf Wuf';  
    },  
};
```

Kreiranje objekta

Možemo kreirati objekt pomoću:

1. literala
2. operatora new - *obradit ćemo kasnije!*
3. Object.create() funkcije - *obradit ćemo kasnije!*

Kreiranje objekta

Možemo kreirati objekt pomoću:

1. literala

```
var cat = {};  
var dog = {  
    breed: 'Golden retriever',  
    bark: function() {  
        return 'Wuf Wuf';  
    },  
    "has toy": false,  
};
```

Svojstva objekta (properties)

Svojstva objekta imaju ime i vrijednost, the opisne attribute.

Vrijednost svojstva može biti bilo koja JS izjava, ili get/set funkcija (ES5).

```
var dog = {  
  breed: 'Golden retriever',  
  bark: function() {  
    return 'Wuf Wuf';  
  },  
}
```

Svojstva objekta, sa opisnim atributima:
writable,
enumerable,
configurable

Svojstva objekta (properties)

Dohvaćanje i postavljanje

Svojstva možemo postaviti i dohvatiti sa točkom (.) ili zagradama ([]).

```
var dog = {};  
dog.breed = 'Golden retriever';  
dog['other breed'] = 'Chihuahua';
```



```
var dog = {  
    breed: 'Golden  
retriever',  
    'other breed':  
    'Chihuahua',  
};
```

```
var firstBreed = dog.breed;  
var secondBreed = dog['second breed'];
```

Svojstva objekta (properties)

Micanje svojstva

Svojstva mičemo sa ključnom riječi **delete**. U slučaju da je micanje uspjelo ili objekt niti nema to svojstvo, izjava vraća vrijednost true.

```
delete dog.breed; // dog objekt više nema svojstvo breed
```

```
var hasDeleted = delete dog['toy'];  
console.log(hasDeleted);
```

Probajte pristupiti, postaviti i micati par svojstava u svom vlastitom objekt literalu!

Metode nad objektom

propertyIsEnumerable	Provjerava da li je svojstvo objekta brojivo (i nije nasljeđeno)
toString / toLocaleString	Pretvara objekt u (lokalizirani) string
valueOf	Pretvara objekt u primitivnu vrijednost
toJSON / JSON.stringify(object)	Pretvara objekt u serializirani JSON objekt (string)

JSON format

JavaScript Object Notation, podskup JavaScript sintakse, vrlo popularan format za prijenos podataka!

JSON sintaksa ima sljedeća pravila:

1. Podaci su u parovima ime / vrijednost
2. Ime vrijednosti mora biti string
3. Stringovi se pišu isključivo sa dvostrukim navodnicima
4. Vrijednost JSON svojstva ne može biti funkcija (bit će zanemarena u pretvorbi)

JSON format - primjer

```
// JavaScript objekt
var person = {
    name: 'John',
    age: 31,
    city: 'New York',
    speak: function(word) {
        return word;
    }
};
```

```
// Pretvoreni JSON objekt
{
    "name": "John",
    "age": 31,
    "city": "New York"
}
```

Osnove programiranja

Objekti - vježba

Objekti - 2.dio - vježba

Pratite upute u Upute.txt

Trajanje vježbe: 45min

Osnove programiranja

Standardni ugrađeni objekti

Tipovi objekata po podrijetlu

1. **Nativni** objekt je objekt ili klasa objekata definirana u ECMAScript specifikaciji (Array, Function, Date ...)
2. **Host** objekt je objekt definiran okruženjem domaćina, u našem slučaju web preglednika (npr. HTMLElement objekt)
3. **Korisnički definiran** objekt je svaki objekt kreiran izvršenjem JavaScript koda.

Globalni objekti

Globalni objekt

Samom globalnom objektu može se pristupiti pomoću **this** operatora u globalnom području (ali samo ako se ne koristi strogi način rada ECMAScript 5; u tom slučaju vraća se nedefiniran). U stvari, globalni se opseg sastoji od svojstava globalnog objekta, uključujući i naslijeđena svojstva, ako ih ima.

Globalni objekti

Odnose se na objekte globalnog opsega.

Izraz "globalni objekti" (ili standardni ugrađeni objekti) ovdje se ne smije miješati s globalnim objektom. Ovdje se globalni objekti.

Date objekt

Core JavaScript uključuje konstruktor **Date ()** za kreiranje objekata koji predstavljaju datume i vremena. Ovi objekti Date imaju metode koje pružaju API za jednostavno izračunavanje datuma. Objekti datuma nisu osnovna vrsta kao što su brojevi.

```
var d = new Date();  
var d = new Date(2018, 11, 24, 10, 33, 30, 0);  
var d = new Date("October 13, 2014 11:13:00");  
var d = new Date(-1000000000000);
```

Math objekt

Pored ovih osnovnih aritmetičkih operatora, JavaScript podržava složenije matematičke operacije kroz skup funkcija i konstanti definiranih kao svojstva objekta Math.

```
Math.PI;  
Math.round(4.4); // 4  
Math.sqrt(64);  
Math.ceil(4.4);  
Math.min(0, 150, 30, 20, -8, -200);
```

Regex

Regular expressions

Redovni izrazi obrasci su koji se koriste za podudaranje kombinacija znakova u tekstu. U JavaScript-u su i redoviti izrazi objekti. Ti se obrasci koriste s `exec` i testnim metodama `RegExp`-a, te sa metodama **match**, **matchAll**, **substitute**, **search** i **split** globalnog String objekta.

```
var re = new RegExp('ab+c');  
var re = /ab+c/;
```

Osnove programiranja - 2.dio

Vježba

Osnove programiranja - 2.dio - vježba

Pratite upute u Upute.text

Trajanje vježbe: 45min