# JavaScript

Je plnohodnotný programovací jazyk, ktorý sa používa v internetových stránkach.

Prvýkrát bol predstavený v roku 1995.

Na vývoji pracovali spoločnosti Netscape a spoločnosť Sun Microsystems. JavaScipt obsahuje v názve slovo Java (objektovo-orientovaný programovací jazyk), okrem názvu však s týmto programovacím jazykom nemá nič spoločné – autori sa rozhodli čerpať z veľkej popularity tohto jazyka.

# Implementácia JavaScript do jazyka HTML

Hlavnou metódou pre vkladanie JavaScriptu do stránok HTML je element *script*. Existujú dva možné spôsoby použitia:

* **Vložením JavaScript kódu priamo do stránky.**

<script type=”text/javascript”>
 function povedzAhoj () {
 alert(“Ahoj”);
 }
</script>

* **Načítavať JavaScript z externého súboru.**

<script type=”text/javascript” src=”example.js”> </script>

<script type=”text/javascript” src=”http://www.blabla.sk/example.js”> </script>

Element script sa umiestňuje do časti head.

<html>
 <head>
 <title> Ukážková kapitola </title>
 <script type=”text/javascript” scr=”priklad1.js”></script>
 <script type=”text/javascript” scr=”priklad2.js”></script>
 </head>
 <body>
 …..
 </body>
</html>

Element script obsahuje nasledúce atribúty:

* charset- nepovinný, znaková sada
* defer-nepovinný, obsah sa odložín dovtedy, kým sa neanalyzuje a zobrazí
 defer=“defer“
* language- zastaralý
* src – nepovinný, označuje externý súbor. Namal by byť ďalši kód medzi elementami script...
* type – povinný, nahrádza language
 type=„text/javascript“

# Základy jazyka

* case sensitive – rozlišovanie veľkých a malých písmenok

test Test

* identifikátor = premennej, funkcie, vlastnosti, argument funkcie
	+ Prvým znakom musí byť písmeno, podtržník, znak dolára

prvaPremenna

pomocna1

\_pomocna2
$pom

* Komentár

//toto je jednoriadkový komentár

/\*
\*
\*Toto je viacriadkový komentár
\*/

* Príkazy- je vhodné ukončovať bodko-čiarkou.

var suma = a + b;
var suma2 = a + b

* Kľúčové slová – majú špeciálny význam, alebo vykonávajú špecifické operácie.
	+ Break, case, else, new, return, var, finally, for, switch, while, …
* Premenné – v jazyku JavaScript sú tzv. “voľne typované” = premenné môžu uchovávať rôzne typy dát.
	+ Premenná sa definuje pomocou kľúčového slova var, za ktorým nasleduje názov premennej (identifikátor):

var suma;
var sprava = “ahoj”;

Premená je vytvorená lokálne v rámci oboru platnosti, kde je definovaná. Napríklad, ak je premenná definovaná vo vnútri nejakej funkcie pomocou operátora var, táto premenná je odstránená okamžite, keď sa funkcia ukončí.

* Dátové typy – JavaScript neumožňuje definovanie vlastného dátového typu
	+ undefined – keď je premenná definovaná var bez inicializacie
	+ Null – prázdna hodnota
	+ Boolean – logická hodnota, True/False
	+ Number – najmenšia hodnota Number.MIN\_VALUE, Number.MAX\_VALUE
	+ String - reťazec
	+ Object – neusporiadaný zoznam dvojíc názov-hodnota
* Riadiace príkazy
	+ Príkaz if

if (i>25) {
 alert(“Číslo je väčšie ako 25”);
} else if (i<25) {
 alert(“Číslo je menšie ako 25);
} else {
 alert(“Číslo je rovné 25“);
}

* Príkaz do-while
	+ Cyklus s podmienkou na konci.

do {
//prikaz
} while (vyraz);

* Príkaz while
	+ Cyklus s podmienkou na začiatku.

while (vyraz) {
//prikaz
}

* Príkaz for
	+ Cyklus s podmienkou na začiatku. Na rozdiel od príkazu while, v cycle inicializujeme premennú.

for(inicializacia;vyraz;vyraz pre koniec cyklu) {
//prikaz
}

* Príkaz for-in
	+ Používa sa k prechádzaniu vlastností objektu

for (vlastnost in vyraz)
//prikaz

* Príkaz break
	+ Príkaz break okamžite ukončí cyklus, takže vykonávanie pokračuje príkazom, ktorý nasleduje za cyklom.

var num = 0;

for (var i = 1; I < 10; i++) {

if ( i % 5 == 0) {

 break;

}
 num++;
}
alert(num);

* Príkaz continue
	+ Príkaz continue ukončí prechod tela cyklu, takže vykonávanie pokračuje od začiatku tela cyklu.

var num = 0;
for (var i = 1; I < 10; i++) {

if ( i % 5 == 0) {

continue;

}
 num++;
}
alert(num);

* Príkaz switch

switch (vyraz) {
 case hodnota: prikaz
 break;
 …
 default: prikaz;

# Funkcie

Funkcie umožňujú zapúzdrenie príkazov, ktoré je možné spúšťať kdekoľvek a kedykoľvek. Funkcie sa v JavaScripte deklarujú pomocou kľúčového slova function, za ktorým nasleduje sada parametrov a telo funkcie.

 function nazovFunkcie(param0, param1, …paramN) {
 //prikazy
 }

 function povedzAhoj(meno, sprava) {
 alert (“Ahoj”+ meno + “, “+ sprava);
 }

Túto funkciu je možné zavolať pomocou názvu funkcie, za ktorou uvedieme v zátvorke argumenty.

 povedzAhoj(“Janko“,“ako sa máš?“);
 povedzAhoj(“Kubko“,“a ty sa máš ako?“);

Funkcia nemusí obsahovať žiadny argument.

function spocitaj(cislo1,cislo2){
 return cislo1+cislo2;
}
var vysledok = spocitaj(10,5);
alert(vysledok);

# Pole

Pole je usporiadaný zoznam dát.

var farby = new Array();
var farby2 = new Array(10);
var farby3= new Array (“biela”,”modra”,”cervena”);

alert(farby3[0]) ;
farby3[2] = “cierna”;

Na zistenie počtu prvkov v poli je možné použiť vlastnosť length, ktorá vždy vráti hodnotu 0 alebo vyššiu.

 var farby3= new Array (“biela”,”modra”,”cervena”);
 alert(farby3.length);

# GeoGebra

GeoGebra podporuje dva typy scriptovacích jazykov:

* GGBScript
* JavaScript

´V appletoch je možňé doprogramovať funkčnosť:

* Klikne na nejaký objekt
* Aktualizuje časť objektu (zmení sa hodnota)
* Načíta sa súbor
* JavaScript listener

Scriptovacie okno sa nachádza v nastaveniach objektu, v záložke „Scripting“.



V GeoGebre môžeme používať špeciálne JavaScript metódy, ktoré patria objektu ggbApplet.

 ggbApplet.nazov\_metody(parameter1,parameter2,...parameterN);

Všetky dostupné metódy, ktoré môžeme pri programovaní objektov použiť sa nachádzajú na stránke [http://www.geogebra.org/wiki/en/Reference:JavaScript](http://www.geogebra.org/wiki/en/Reference%3AJavaScript).

Napríklad, nasledujúci script vytvorí 10 bodov A0 až A9 s náhodnými pozíciami.

 for (var i=0; I < 10; i++) {
 ggbApplet.evalCommand(“A\_“+i+“=(random()\*10, random()\*10)“);
 }

Global JavaScript

**Stupeň vrcholu**

Na obrázku sa nachádza applet stupeň vrcholu



Obrázok 18 Applet stupeň vrcholu.

Po kliknutí na hocijaký vrchol v grafickom okne, applet zvýrazní hrany, ktoré smerujú z tohto vrcholu a vypíše sa text *deg(bod) = pocet.*

Algoritmus tohto appletu je pomerne jednoduchý. Na začiatku si vytvoríme pomocnú premennú, ktorú nazveme *pocetHran*. Táto premenná nám uchováva hodnotu koľko hrán smeruje z tohto bodu. Na začiatku musí byť táto premenná inicializovaná na hodnotu 0:

*var pocetHran = 0*

Hrany v GeoGebre majú predvolené názvy – malé písmená v abecede. Preto si vytvoríme pomocnú premennú, ktorú pomenujeme *pomAbeceda* a inicializujeme ju na hodnotu malých písmen abecedy:

*var pomAbeceda = "abcdefghijklmnoprstuvwxyz"*

Nakoľko nevieme dopredu určiť, koľko bude vrcholov a hrán v grafe, použijeme cyklus s pevným počtom opakovaní. Pre každý znak v premennej *pomAbeceda* sa vykonajú nasledujúce príkazy:

*for(var i=0; i<pomAbeceda.length;i++) {*

*var pomZnak = pomAbeceda.charAt(i)*

*if (ggbApplet.exists(pomZnak)) {*

*var pomSegment=ggbApplet.getDefinitionString(pomZnak)*

*if( pomSegment.indexOf(aktualnyObjekt) >= 0){*

*ggbApplet.setLineThickness(pomZnak,6)*

*pocetHran++*

*} else {*

*ggbApplet.setLineThickness(pomZnak,1)*

*}*

*}*

*}*

Do premennej *pomZnak* sa uloží aktuálny znak z premennej *pomAbeceda*. Prvý znak je písmeno *a*, nasleduje *b*, atď. Posledný cyklus prebehne s písmenom *z*. Pomocou GeoGebra metódy *exists()* zistíme, či dané písmeno (vrchol) existuje v grafickom okne appletu. Syntax metódy:

*boolean exists(String objName)*

Ak objekt existuje, tak do premennej *pomSegment* uložíme predvolenú hodnotu definície hrany (napríklad pre hranu *AB* je predvolená definícia *Segment[A,B]* ).

Hodnotu definície hrany získame pomocou metódy:

*String getDefinitionString(String objName)*

Následne pomocou JavaScript metódy *indexOf()* zistíme, či sa v definícii hrany nachádza vrchol, na ktorý sme naposledy klikli v grafickom okne. Ak sa tam tento vrchol nachádza, nastavíme hrúbku tejto hrany na hodnotu 6. Hrúbku čiary nastavujeme pomocou GeoGebra metódy *setLineThickness():*

*void setLineThickness(String objName, int thickness)*

Hrúbka čiary môže mať hodnotu od 0 do 13. V tejto podmienke zvýšime hodnotu premennej *pocetHran* o hodnotu 1. Ak sa vrchol, na ktorý sme naposledy klikli nenachádza v definícií hrany, tak nastavíme hrúbku tejto hrany na hodnotu 1. Poslednú vec, čo musíme ešte urobiť je vypísať text – koľko hrán smeruje z aktuálneho vrcholu. Pomocou metódy *setTextValue()* nastavíme pre objekt s názvom *text1* text *deg(bod) = pocet.*

*ggbApplet.setTextValue("text1","deg("+aktualnyObjekt+") ="+pocetHran)*

###

### Publikácia appletu

Materiály vytvorené v programe GeoGebra je možné ďalej šíriť. V časti menu
*Súbor 🡪Exportovať* je možné vybrať si z nasledujúcich možností:

* uložiť ako súbor Geogebra (prípona gbb),
* uložiť ako stránku HTML,
* exportovať ako obrázok PNG, PDF, EPS, SVG a EMF,
* exportovať ako animovaný obrázok GIF.