WEB ACADEMY

Front End за Начинаещи JavaScript

Автор: Йордан Енев

http://WebAcademy.bg/

RegExp обект

RegExp обектът служи за работа с регулярни изрази. Регулярният израз е поредица от символи, която образува т. нар. модел за търсене. Този модел за търсене може да се използва в операции за търсене и заместване на текст.

Пример за лесен RegExp

```
var str = "Visit W3Schools!";
var res = str.replace(/w3schools/i, "WebAcademy");
var n   = str.search(/w3s/i); // n = 6; Броенето започва от нула :-)
```

- Синтаксис: */регулярен израз (pattern)/модификатори*
- Методи за работа с RegExp:

Method	Description
exec()	Tests for a match in a string. Returns the first match
test()	Tests for a match in a string. Returns true or false
toString()	Returns the string value of the regular expression

RegExp обект

- Модификатори:
  - Служат за задаване на различни опции
  - Списък с често използвани модификатори

Modifier	Description
i	Perform case-insensitive matching
g	Perform a global match (find all matches rather than stopping after the first match)
m	Perform multiline matching

- Набор от символи

Expression	Description
[abc]	Find any of the characters between the brackets
[0-9]	Find any of the digits between the brackets
(x|y)	Find any of the alternatives separated with |

  - символа ^ задава отрицание

RegExp обект

- Метасимволи:
  - Представляват символи със специално значение
  - Списък с често използвани метасимволи (вижте всички)

Metacharacter	Description
.	Find a single character, except newline or line terminator
\w	Find a word character
\d	Find a digit
\s	Find a whitespace character
\b	Find a match at the beginning/end of a word
\0	Find a NUL character
\n	Find a new line character
\f	Find a form feed character
\r	Find a carriage return character
\t	Find a tab character

  - Главните букви указват отрицание. Например \D - не цифра

RegExp обект

- Модифициране на количество
  - Задава се чрез така наречените quantifiers
  - Списък с всички модификатори на количество

Quantifier	Description
n+	Matches any string that contains at least one *n*
n*	Matches any string that contains zero or more occurrences of *n*
n?	Matches any string that contains zero or one occurrences of *n*
n{X}	Matches any string that contains a sequence of *X* *n*'s
n{X,Y}	Matches any string that contains a sequence of X to Y *n*'s
n{X,}	Matches any string that contains a sequence of at least X *n*'s
n$	Matches any string with *n* at the end of it
^n	Matches any string with *n* at the beginning of it
?=n	Matches any string that is followed by a specific string *n*
?!n	Matches any string that is not followed by a specific string *n*

Събития

- Събитията (events) възникват, когато нещо се случи, а JavaScript от своя страна реагира на тези събития
- Свързват се с конкретен елемент от страницата

```
<button onclick="this.innerHTML=Date()">The time is?</button>
```

- Най-често използвани събития:

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

DOM
Document Object Model

HTML DOM & CSS

Какво ни е нужно?

- Намиране на HTML елементи
- Промяна на HTML елементи
- Добавяне и премахване на елементи

Вижте още:

- JavaScript HTML DOM Document
- JavaScript HTML DOM Elements
- CSS-JS References

DOM
Document Object Model

HTML DOM & CSS

Намиране на HTML елементи

- по id атрибут
- по име на таг
- по име на клас
- по CSS селектор
- по колекция от обекти

```
var myElement = document.getElementById("main");
var x = document.getElementsByTagName("p");
var y = document.getElementsByClassName("intro");
var z = document.querySelectorAll("p.intro");
```

```
var x = document.forms["frm1"];
var text = "";
var i;
for (i = 0; i < x.length; i++) {
    text += x.elements[i].value + "<br>";
}
document.getElementById("demo").innerHTML = text;
```

First name: Donald
Last name: Duck

Submit

See Result

Donald
Duck
Submit

DOM
Document Object Model

HTML DOM & CSS

Промяна на HTML елементи е възможна чрез:

- HTML съдържание
- име на атрибут
- метод setAttribute
- свойство style

```
element.innerHTML =  new html content;
element.attribute = new value;
element.setAttribute(attribute, value);
element.style.property = new style; // Псевдо код
```

```
<div id="main"></div>                                    // Примерен код
<script>
   var element = document.getElementById('main');
   element.innerHTML =  "new html content";
   element.title = "Change Title Attribute";
   element.setAttribute("title", "New Title Again");
   element.style.border = "1px solid blue";
</script>
```

DOM
Document Object Model

HTML DOM & CSS

Методи за добавяне и премахване на елементи:

- document.createElement(element) - създава елемент
- document.removeChild(element) - премахва елемент
- document.appendChild(element) - добавя елемент
- document.replaceChild(element) - замества елемент

```
<input type="button" onclick="addNew();" value="+"><ol id="my_list"></ol>
<script>
var i = 1;
function addNew(){
    var element = document.createElement('li');
    element.id= "item_"+i;
    element.innerHTML = 'New Item '+i;
    document.getElementById('my_list').appendChild(element);  i++;
}</script> <!-- Динамично добавяне на нов елемент към списък -->
```

CSS - JS References

CSS Property	JavaScript Reference
background	background
background-attachment	backgroundAttachment
background-color	backgroundColor
background-image	backgroundImage
background-position	backgroundPosition
background-repeat	backgroundRepeat
border	border
border-bottom	borderBottom
border-bottom-color	borderBottomColor
border-bottom-style	borderBottomStyle
border-bottom-width	borderBottomWidth
border-color	borderColor
border-left	borderLeft
border-left-color	borderLeftColor
border-left-style	borderLeftStyle
border-left-width	borderLeftWidth
border-right	borderRight
border-right-color	borderRightColor
border-right-style	borderRightStyle

CSS Property	JavaScript Reference
border-right-width	borderRightWidth
border-style	borderStyle
border-top	borderTop
border-top-color	borderTopColor
border-top-style	borderTopStyle
border-top-width	borderTopWidth
border-width	borderWidth
clear	clear
clip	clip
color	color
cursor	cursor
display	display
filter	filter
font	font
font-family	fontFamily
font-size	fontSize
font-variant	fontVariant
font-weight	fontWeight
height	height

CSS - JS References

CSS Property	JavaScript Reference
left	left
letter-spacing	letterSpacing
line-height	lineHeight
list-style	listStyle
list-style-image	listStyleImage
list-style-position	listStylePosition
list-style-type	listStyleType
margin	margin
margin-bottom	marginBottom
margin-left	marginLeft
margin-right	marginRight
margin-top	marginTop
overflow	overflow
padding	padding
padding-bottom	paddingBottom
padding-left	paddingLeft
padding-right	paddingRight
padding-top	paddingTop
page-break-after	pageBreakAfter

CSS Property	JavaScript Reference
page-break-before	pageBreakBefore
position	position
float	cssFloat
text-align	textAlign
text-decoration	textDecoration
text-decoration: blink	textDecorationBlink
text-decoration: line-through	textDecorationLineThrough
text-decoration: none	textDecorationNone
text-decoration: overline	textDecorationOverline
text-decoration: underline	textDecorationUnderline
text-indent	textIndent
text-transform	textTransform
top	top
vertical-align	verticalAlign
visibility	visibility
width	width
z-index	zIndex

Задача 1

- Чрез JavaScript код генерирайте шахматна дъска.

Жокер:

- За визуално представяне на дъската можете да ползвате както дивове, така и таблици

От какво се нуждаете?

- 1-2 цикъла For
- доза логическо мислене
- щипка старание
- увереност при делението по модул :-)
- 5-10 минути върху клавиатурата

Задача 2

- Направете "команден пулт" за управление на "совалка"

Жокер:

- Нека "совалката" да е изображение.
- Направете 4 бутона с които да движите "совалката"
- Совалката се движи чрез смяна на позицията си

От какво се нуждаете?

- 1 изображение със совалка
- 4-5 бутона
- доза опит с координатни системи
- щипка старание + свидетелство за управление на НЛО :-)
- 15-20 минути върху клавиатурата

Задача 3

- Направете форма за добавяне на готварска рецепта.

Жокер:

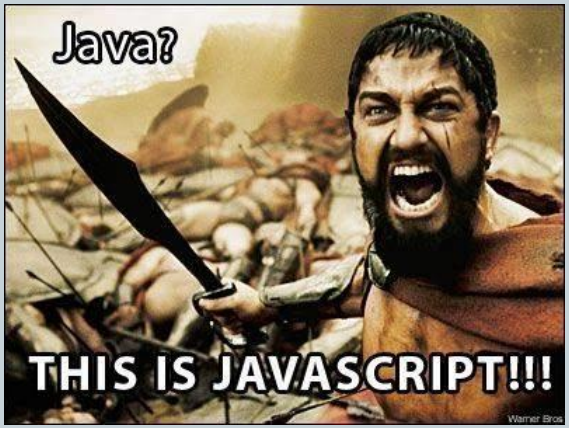
- Продуктите трябва да могат да се добавят автоматично
- За всеки продукт трябва да имаме име и количество

От какво се нуждаете?

- 1 текстово поле за кратък текст (заглавие)
- 1 текстово поле за дълъг текст (описание)
- 1 таблица с необходими продукти
- опит в готварството и смелост да пишете append
- 50-60 минути печене на бавен огън върху клавиатурата

ВЪПРОСИ

Поздравления!



Този модул приключи!

Желаем Ви късмет с финалния изпит!

Вече знаете, какво е HTML & CSS & JS!

HTML = How To Meet Ladies
CSS = Countless Sex Styles
JS = Just Sex