

Changing HTML Elements

Property	Description
<code>element.innerHTML = new html content</code>	Change the inner HTML of an element
<code>element.attribute = new value</code>	Change the attribute value of an HTML element
<code>element.style.property = new style</code>	Change the style of an HTML element
Method	Description
<code>element.setAttribute(attribute, value)</code>	Change the attribute value of an HTML element

Adding and Deleting Elements

Method	Description
<code>document.createElement(element)</code>	Create an HTML element
<code>document.removeChild(element)</code>	Remove an HTML element
<code>document.appendChild(element)</code>	Add an HTML element
<code>document.replaceChild(new, old)</code>	Replace an HTML element
<code>document.write(text)</code>	Write into the HTML output stream

Adding Events Handlers

Method	Description
<code>document.getElementById(id).onclick = function(){code}</code>	Adding event handler code to an onclick event

Figure 1Пример клик на било каков елемент, div, textarea, го земаме неговото ID и ставаме некаква функција на него, како во задачата со коцките, клик на div смени radius...

Finding HTML Elements by Tag Name

This example finds all `<p>` elements:

Example

```
const element = document.getElementsByTagName("p");
```

document.body	Returns the <body> element
document.cookie	Returns the document's cookie
document.doctype	Returns the document's doctype
document.documentElement	Returns the <html> element
document.documentMode	Returns the mode used by the browser
document.documentURI	Returns the URI of the document
document.domain	Returns the domain name of the document server
document.domConfig	Obsolete.
document.embeds	Returns all <embed> elements
document.forms	Returns all <form> elements
document.head	Returns the <head> element
document.images	Returns all elements
document.implementation	Returns the DOM implementation
document.inputEncoding	Returns the document's encoding (character set)
document.lastModified	Returns the date and time the document was updated
document.links	Returns all <area> and <a> elements that have a href attribute

If you want to find all HTML elements that match a specified CSS selector (id, class names, types, attributes, values of attributes, etc), use the `querySelectorAll()` method.

This example returns a list of all `<p>` elements with `class="intro"`.

Example

```
const x = document.querySelectorAll("p.intro");
```

To change the value of an HTML attribute, use this syntax:

```
document.getElementById(id).attribute = new value
```

This example changes the value of the src attribute of an `` element:

Example

```
<!DOCTYPE html>
<html>
<body>



<script>
document.getElementById("myImage").src = "landscape.jpg";
</script>
```

Слика 1 На местото од .attribute МЕХУБАЈ кој ампубым ти треба, src, href...

JavaScript Form Validation

HTML form validation can be done by JavaScript.

If a form field (fname) is empty, this function alerts a message, and returns false, to prevent the form from being submitted:

JavaScript Example

```
function validateForm() {  
  let x = document.forms["myForm"]["fname"].value;  
  if (x == "") {  
    alert("Name must be filled out");  
    return false;  
  }  
}
```

If a form field (fname) is empty, the **required** attribute prevents this form from being submitted:

HTML Form Example

```
<form action="/action_page.php" method="post">  
  <input type="text" name="fname" required>  
  <input type="submit" value="Submit">  
</form>
```

Слика 2 Во form кај полето за внесување име ако пишам required НЕМА ДА ДОЗВОЛИ ДА СЕ ИСПРАТИ Form-от без да биде пополнето!!!

Constraint Validation CSS Pseudo Selectors

Selector	Description
:disabled	Selects input elements with the "disabled" attribute specified
:invalid	Selects input elements with invalid values
:optional	Selects input elements with no "required" attribute specified
:required	Selects input elements with the "required" attribute specified
:valid	Selects input elements with valid values

To change the style of an HTML element, use this syntax:

```
document.getElementById(id).style.property = new style
```

Слика 3 НА МЕСТОТО ОД property пишуваш пример, backgroundColor, color, border...

```

<!DOCTYPE html>
<html>
<body>

<h2>JavaScript addEventListener()</h2>

<p>This example uses the addEventListener() method to add many events on the same button.</p>

<button id="myBtn">Try it</button>

<p id="demo"></p>

<script>
var x = document.getElementById("myBtn");
x.addEventListener("mouseover", myFunction);
x.addEventListener("click", mySecondFunction);
x.addEventListener("mouseout", myThirdFunction);

function myFunction() {
  document.getElementById("demo").innerHTML += "Moused over!<br>";
}

function mySecondFunction() {
  document.getElementById("demo").innerHTML += "Clicked!<br>";
}

function myThirdFunction() {
  document.getElementById("demo").innerHTML += "Moused out!<br>";
}
</script>

</body>
</html>

```

JavaScript addEventListener()

This example uses the addEventListener() method to add many events on the same button.

Try it

Moused over!
Moused out!
Moused over!
Moused out!
Moused over!
Moused out!
Clicked!
Moused out!

The removeEventListener() method

The `removeEventListener()` method removes event handlers that have been attached with the `addEventListener()` method:

Example

```
element.removeEventListener("mousemove", myFunction);
```

Navigating Between Nodes

You can use the following node properties to navigate between nodes with JavaScript:

- `parentNode`
- `childNodes[nodenum]`
- `firstChild`
- `lastChild`
- `nextSibling`
- `previousSibling`

Accessing the `innerHTML` property is the same as accessing the `nodeValue` of the first child:

```
myTitle = document.getElementById("demo").firstChild.nodeValue;
```

Accessing the first child can also be done like this:

```
myTitle = document.getElementById("demo").childNodes[0].nodeValue;
```

Example

Change the text color of all <p> elements:

```
const myCollection = document.getElementsByTagName("p");
for (let i = 0; i < myCollection.length; i++) {
  myCollection[i].style.color = "red";
}
```

Example

```
const myNodeList = document.querySelectorAll("p");
```

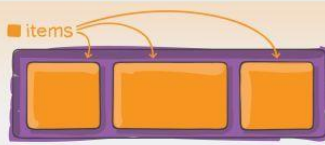
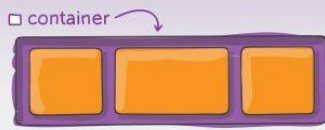
The elements in the NodeList can be accessed by an index number.

To access the second <p> node you can write:

```
myNodeList[1]
```

CSS Flexbox

a guide from
* CSS-TRICKS



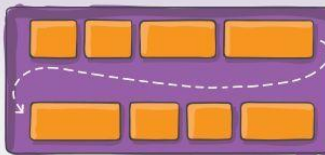
```
.container {
  display: flex; /* or inline-flex */
}
```

flex-direction



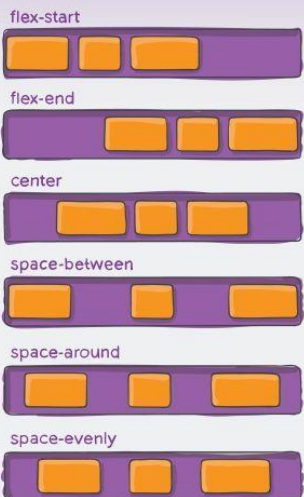
```
.container {
  flex-direction: row | row-reverse |
  column | column-reverse;
}
```

flex-wrap



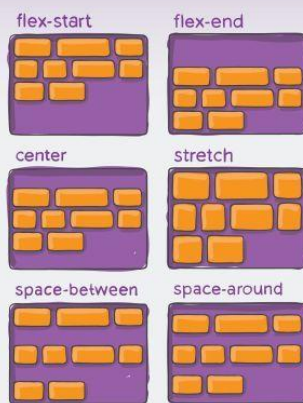
```
.container {
  flex-wrap: nowrap | wrap | wrap-reverse;
}
```

justify-content



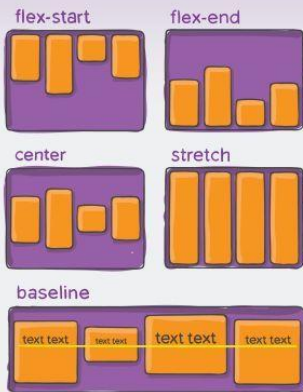
```
.container {
  justify-content: flex-start | flex-end |
  center | space-between | space-around |
  space-evenly;
}
```

align-content



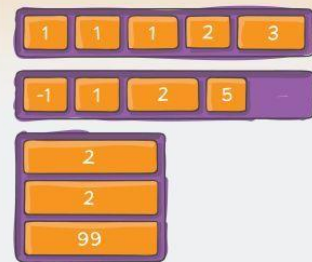
```
.container {
  align-content: flex-start | flex-end |
  center | space-between | space-around |
  space-evenly | stretch;
}
```

align-items



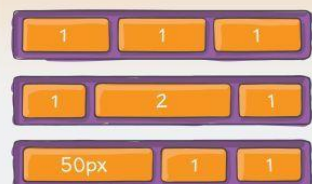
```
.container {
  align-items: stretch | flex-start |
  flex-end | center | baseline;
}
```

order



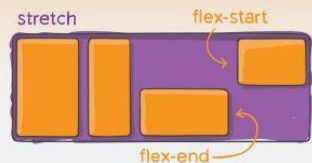
```
.item {
  order: 5; /* default is 0 */
}
```

flex-shrink, flex-grow, flex-basis



```
.item {
  flex-shrink: 1; /* default is 1 */
  flex-grow: 2; /* default is 0 */
  flex-basis: 50px; /* default auto */
}
```

align-self



```
.item {
  align-self: auto | flex-start |
  flex-end | center | baseline | stretch;
}
```


Прв парцијален испит

Основни елементи од HTML, CSS и JavaScript

OWEB, Copyright © 2020

Пополнете ја следнава форма:

Лични информации:

Име:

Презиме:

Кориснички податоци:

Email:

Password:

Пополнете ги и следниве детали:

Адреса:

Улица и број

Телефон:

000/000-000

OWEB, Copyright © 2020

This page says

Product peach not in shop.

OK

Quantity

Unit Price

Add item

OWEB, Copyright © 2020