Univerzitet Singidunum

Web Programming

www.singidunum.ac.rs

Form elements

- Every HTML document loaded inside browser (including page elements which are invisible to user) become **document** object
- In JavaScript, object document refers to the complete page where this code is activated, and can be used to call each element inside script
- This is the DOM organization (Document Object Model)
- Page is represented as set of all elements (tags, images, text, links etc.) and document contains them all

Form elements

- User can interact with the site in two ways
  1. Through form elements, this being the only way to enter the content
  2. Through events – we can detect characteristic activities of the user, such as click, mouse movement over some area, page loading…
- Both can be implemented with JavaScript

Form elements

- Every form has elements where user can provide some data
- Goal is to get this data, validate it and at the end process it
- This can also be done with the server language, but it is better to check on the client if everything is according to predefined format, and then send it to the server
- First step is to connect HTML form with script language, and assign those values to some variables inside script language, and further process it inside script language
- Validation is done through regular expressions
  - One of the most important Java Script tasks

Form elements

- Any element in the form can be accessed in two ways:
  - HTML id attribute

    var x=document.getElementById(‘element_id’)
  - HTML name attribute

    var y=document.form_name.element_name
- In this way element is selected, and we can apply and change certain properties (name, value, length, selected, src, alt…)

Form elements

- So the first step in content management is to access document object
- Second step is to select concrete element of the form by name or id attribute
- Third step is to define property which we want to access
- For example, to get content that user entered:

  1. With HTML attribute id

  – var x=document.getElementById(‘element_id’).value

  2. With HTML attribute name

  – var y=document.form_name.element_name.value

Form elements

- Two ways for assigning content, like in all programming languages (similar to get() and set() methods)

1. From the form element to the variable

```
var x=document.getElementById(‘element_id’).value;
```

2. From the variable to the form element

```
document.getElementById(‘element_id’).value = x;
```

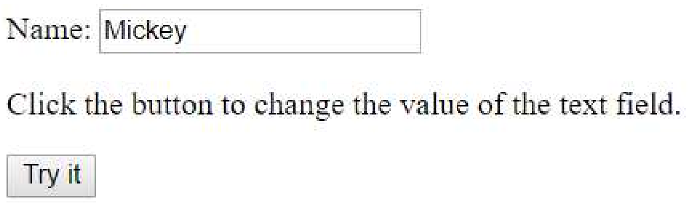
Form elements

- With document object, we can access to any of its elements
- Usually it is tag, tag attribute and similar

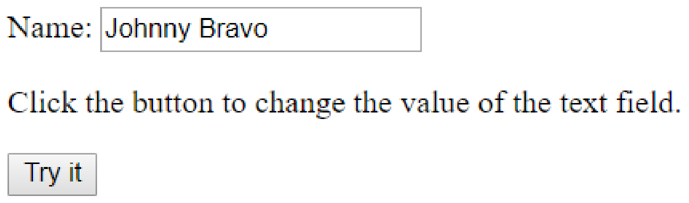
- Accessing tag is possible also with:
  ```
  var x=document.getElementByTagName(‘name_of_tag’);
  ```
- It is possible to get value of tag attribute with:
  ```
  var x=document.getElementById(‘element_id’).src;
  ```
- To change the value of tag attribute:
  ```
  document.getElementById(‘element_id’).src = “1.jpg”;
  ```

Form elements

```
<html>
<body>

Name: <input type="text" id="myText" value="Mickey">

<p>Click the button to change the value of the text field.</p>

<button onclick="myFunction()">Try it</button>

<script>
function myFunction() {
    document.getElementById("myText").value = "Johnny Bravo";
}
</script>

</body>
</html>
```


Name: Mickey

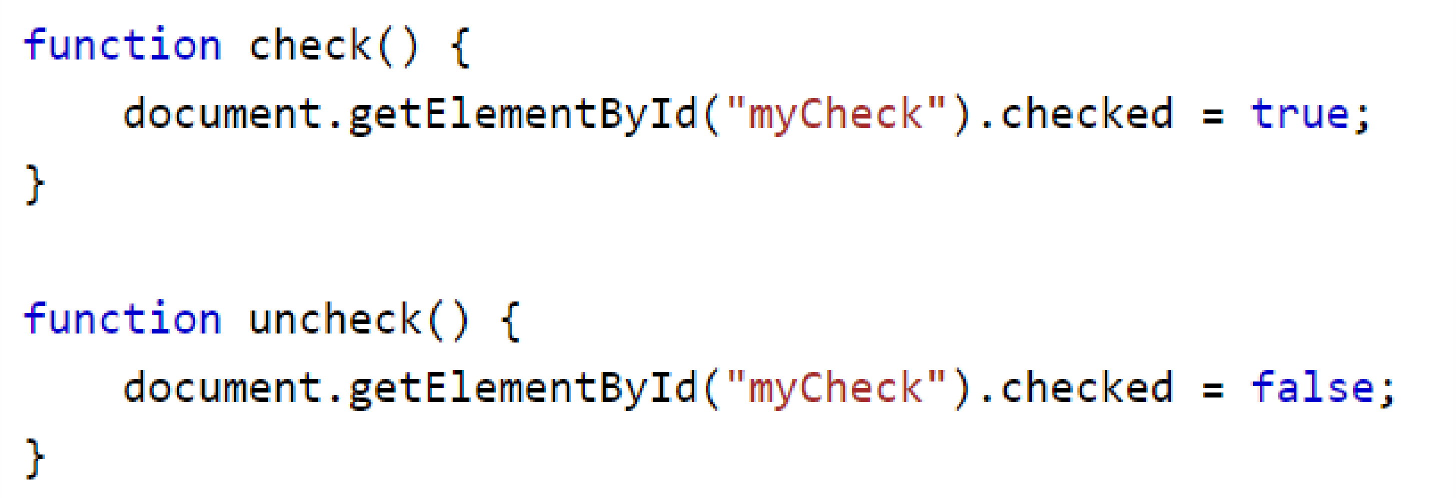
Click the button to change the value of the text field.


Name: Johnny Bravo

Click the button to change the value of the text field.

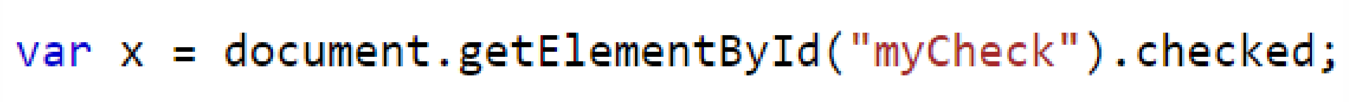
Form elements

- Input Checkbox checked Property
- Set the checked state of a checkbox:



```
function check() {
    document.getElementById("myCheck").checked = true;
}

function uncheck() {
    document.getElementById("myCheck").checked = false;
}
```

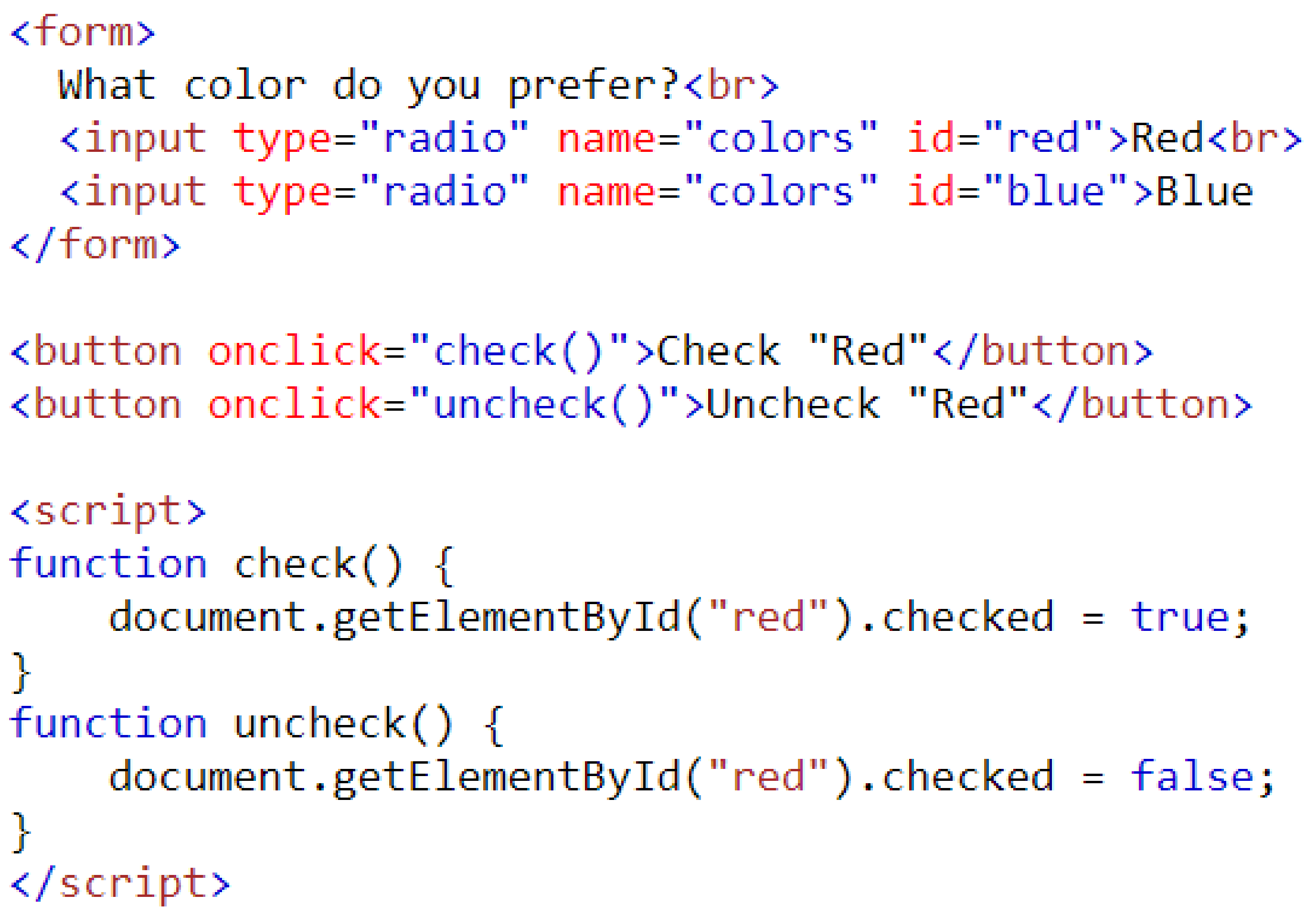
- Find out if a checkbox is checked or not:



```
var x = document.getElementById("myCheck").checked;
```

Form elements

- Input Radio checked Property



```
<form>
  What color do you prefer?<br>
  <input type="radio" name="colors" id="red">Red<br>
  <input type="radio" name="colors" id="blue">Blue
</form>

<button onclick="check()">Check "Red"</button>
<button onclick="uncheck()">Uncheck "Red"</button>

<script>
function check() {
    document.getElementById("red").checked = true;
}
function uncheck() {
    document.getElementById("red").checked = false;
}
</script>
```

Arrays

- Important structure, used every time there is more than one variable
- Syntax in JavaScript is similar to other programming languages
- To create an array, it is not necessary to explicitly define the variable type
- Key word **Array**
- Indexing starts at 0
- Access the element with ArrayName[position]

Arrays

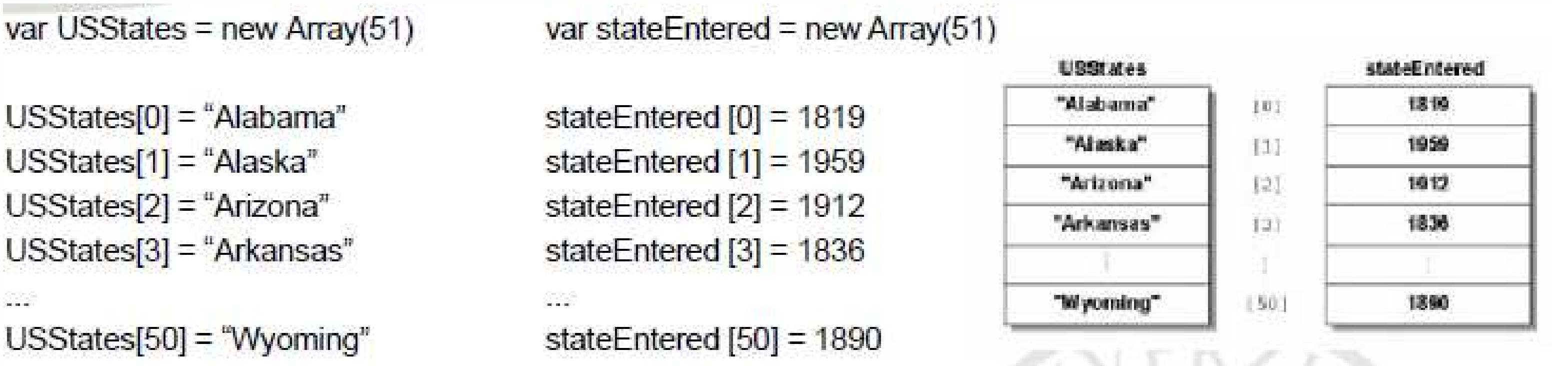
- To create an array

```
var prices = new Array(100);
```

- To write into the array:

```
array[5] = 200.33
```

Arrays



```
var USStates = new Array(51)

USStates[0] = “Alabama”
USStates[1] = “Alaska”
USStates[2] = “Arizona”
USStates[3] = “Arkansas”
...
USStates[50] = “Wyoming”
```

```
var stateEntered = new Array(51)

stateEntered [0] = 1819
stateEntered [1] = 1959
stateEntered [2] = 1912
stateEntered [3] = 1836
...
stateEntered [50] = 1890
```

USStates		stateEntered
"Alabama"	[0]	1819
"Alaska"	[1]	1959
"Arizona"	[2]	1912
"Arkansas"	[3]	1836
⋮	⋮	⋮
"Wyoming"	[50]	1890

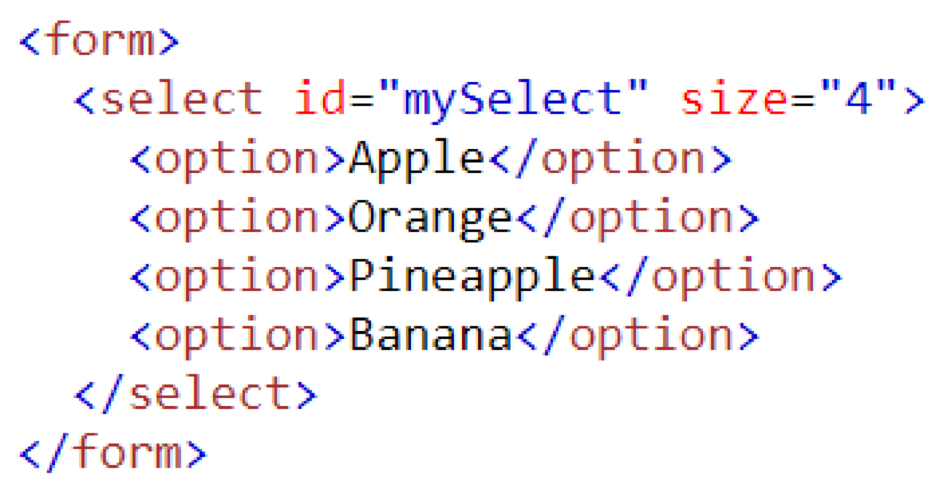
```
function searchNumberForEnteredState()
{
var chosenstate = document. Form.inputstate.value
for ( var i = 0; i < USStates.length; i++)
{
if (USStates[i] == chosenstate) {Break;}
}
alert(“For state “ + chosen state + ”number is ” + stateEntered[i] )
}
```

Arrays and forms

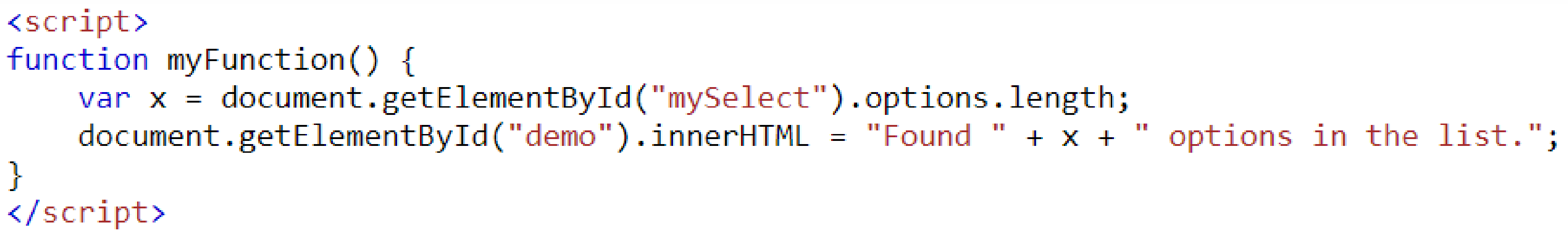
- A large number of uses of arrays when working with forms
- Most often with processing data from drop down lists, radio group or check boxes
- Array is necessary when structure received from the web page is in the form of array, for example drop down list
- In these cases, name of the array is predefined, for example options

Form elements

- Control of the select type (drop down list)

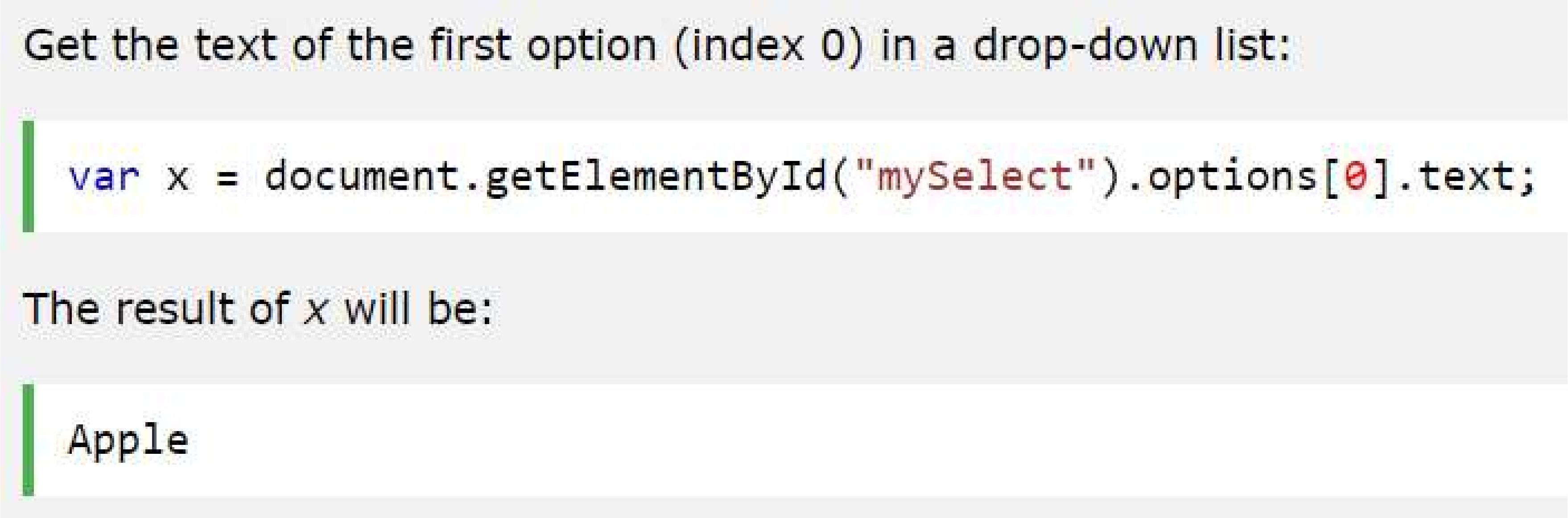


```
<form>
  <select id="mySelect" size="4">
    <option>Apple</option>
    <option>Orange</option>
    <option>Pineapple</option>
    <option>Banana</option>
  </select>
</form>
```



```
<script>
function myFunction() {
    var x = document.getElementById("mySelect").options.length;
    document.getElementById("demo").innerHTML = "Found " + x + " options in the list.";
}
</script>
```

Form elements



Get the text of the first option (index 0) in a drop-down list:

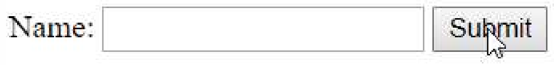
```
var x = document.getElementById("mySelect").options[0].text;
```

The result of *x* will be:

```
Apple
```

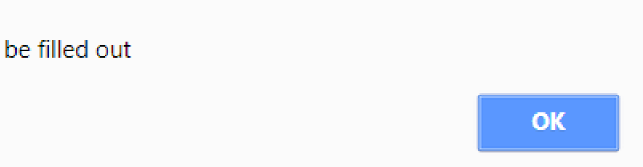
Form elements

- Basic validation (just checking if the field is filled), regex validation will come later



Name: Submit

```
<html>
<head>
<script>
function validateForm() {
    var x = document.forms["myForm"]["fname"].value;
    if (x == "") {
        alert("Name must be filled out");
        return false;
    }
}
</script>
</head>
<body>

<form name="myForm" action="/action_page.php"
onsubmit="return validateForm()" method="post">
Name: <input type="text" name="fname">
<input type="submit" value="Submit">
</form>

</body>
</html>
```



Name must be filled out

Opening / closing new window

- New window can be created to open some page in HTML code
- New window size can be controlled with setting height and width, or can be closed by name:

```
function openNewWindow(){
var mywindow = window.open("1.html",
"height=300, width=200")}
function closeNewWindow(){
mywindow.close()}
```