Functions as Objects

Dr Enrico Gerding
JavaScript Functions

• So far we have used and created basic functions
• However, in JavaScript functions are also objects, just as arrays and other variables
• This means functions can be passed on as arguments to other functions
• This allows for very powerful, but often complex programming, but is an essential part of how JavaScript is used e.g. for handling events
Defining a function

```javascript
function (arguments) functionName { 
    statement(s)
}
```

Or directly as an object:

```javascript
var functionName = function (arguments) { 
    statement(s)
}
```
Example

```javascript
var myFunc;

// toss a coin and choose the behaviour of myFunc
if (Math.random() < 0.5) {
    myFunc = function() {
        alert("heads");
    };
} else {
    myFunc = function(elem) {
        alert("tails");
    };
}
myFunc();
```
Example with arguments

```javascript
var myFunc;

// toss a coin and choose the behaviour of myFunc
if (Math.random() < 0.5) {
    myFunc = function(elem) {
        elem.innerHTML="heads";
    };
} else {
    myFunc = function(elem) {
        elem.innerHTML="tails";
    };
}
myFunc(document.getElementById("result"));
```
Variables within scope can also be used

```javascript
var value=Math.random();

var myFunc = function(elem) {
    elem.innerHTML="The value is:"+value;
};

myFunc(document.getElementById("result"));
```
Used e.g. to add event listeners in JavaScript

- Often we want to add or remove events directly in JavaScript
- To do so, we have to specify the event but also the function that is called, the so-called *listener function*

**Syntax:**

```javascript
element.addEventListener(event, function);
```

```javascript
element.removeEventListener(event, function);
```
Example 1:

```html
<h1 id="myElement">Hello World</h1>
<script>
    function doSomething() {
        alert("hello!");
    }
    document.getElementById("myElement").addEventListener("click", doSomething);
</script>
```
Example 2:

```html
<h1 id="myElement">Hello World</h1>
<script>
  var soSomething = function() {
    alert("hello!"瘙;
  }

  document.getElementById("myElement ").addEventListener("click", doSomethin
g);
</script>
```
Example 3: Anonymous form

```html
<h1 id="myElement">Hello World</h1>
<script>
    document.getElementById("myElement").addEventListener("click", function() {
        alert("hello!");
    });
</script>
```
Example 4: variable scope

<h1 id="myElement">Hello World</h1>
<script>
    var name="Enrico";
    document.getElementById("myElement").addEventListener("click",
    function() {
        alert("hello " + name);
    });
</script>
Note regarding events

- Events can only be attached to DOM elements which have already been loaded
- To make sure the HTML page had loaded, use the `onload` event in the `body` tag

```html
<body onload="doSomething()">
...
</body>
```
Conclusion

• Functions in JavaScript are objects
• Can be used as arguments to other functions just as any other object
• Anonymous functions are often used