

Unobtrusive JavaScript

Separation of Web

- Content (HTML) - what is it?
- Presentation (CSS) - how does it look?
- Behavior (JavaScript) - how does it respond to user interaction?

```
<button id="ok" onclick="okayClick();">OK</button>  
// called when OK button is clicked  
function okayClick() {  
    alert("booyah");  
}
```



This is Bad (JS Inside HTML)

When does JS Code runs

Your file's JS code runs the moment the browser loads the script tag

- Any variables are declared immediately
- Any functions are declared but not called, unless your global code explicitly calls them

```
<head>
  <script src="myfile.js" type="text/javascript"></script>
</head>

<body> ... </body>
```

// global code

```
var x = 3;
function f(n) { return n + 1; }
function g(n) { return n - 1; }
x = f(x);
```

//global code

Problem

Problem: global JS code runs the moment the script is loaded

Script in head is processed before page's body has loaded

- No elements are available yet or can be accessed yet via the DOM
- We need a way to attach the handler after the page has loaded...

Will not work // page is not ready

```
<script>  
    function myfunc(){  
        console.log("btn clicked");  
    }  
    document.getElementById("myBtn").onclick = myfunc;  
</script>  
  
<button id="myBtn">Click me</button>
```

```
// this will run once the page has finished loading
function functionName() {
    element.event = functionName;
    element.event = functionName;
    ...
}

window.onload = functionName; // global code
```

JS

window.onload event

This will work

```
<script>
    function myfunc(){
        console.log("btn clicked");
    }
    function doBindings(){
        document.getElementById("myBtn").onclick = myfunc;
    }
    window.onload = doBindings;
</script>
<button id="myBtn">Click me</button>
```

Anonymous Functions

JavaScript allows you to declare anonymous functions

Quickly creates a function without giving it a name

Can be stored as a variable, attached as an event handler,
etc.

```
function (parameters) {  
    statements;  
}
```

JS

Why it worked

Delayed Execution of code.

Binding Code is run right after the DOM is ready

Make Code Short

```
<script>
function myfunc(){
    console.log("btn clicked");
}
window.onload = function(){
    document.getElementById("myBtn").onclick = myfunc;
};
</script>
```

More Compact

```
<script>
window.onload = function () {
document.getElementById("myBtn").onclick =
function () {
    console.log("btn clicked");
};
};
</script>
```

```
window.onload = function() {
    var okButton = document.getElementById("ok");
    okButton.onclick = okayClick;
};

function okayClick() {
    alert("booyah");
}
```

JS

OK

output

Anonymous Function Example

Common Mistakes

```
window.onLoad = pageLoad;  
window.onload = pageLoad;
```

JS

```
window.onload = pageLoad();  
window.onload = pageLoad;  
  
okButton.onclick = okayClick();  
okButton.onclick = okayClick;
```

JS

Form Validation

```
<form id="gform" action="https://google.com/search">  
  <input id="query" type="text" name="q" value="cui"/>  
  <span id="error-message" style="display: none;">Query is Required </span>  
  <input type="submit">  
</form>
```

Binding

```
window.onload = function(){
    var gform = document.getElementById("gform");
    gform.onsubmit = handleFormSubmit;
}
```

Form Validation Handler

```
function handleFormSubmit(event){  
    let query = document.getElementById("query");  
    if(query.value){  
        console.log("valid form")  
    }else {  
        console.log("Invalid Form");  
        query.classList.add("error");  
        let errorMessage = document.getElementById("error-message");  
        errorMessage.style.display = "inline";  
        event.preventDefault();  
    }  
}
```