JavaScript

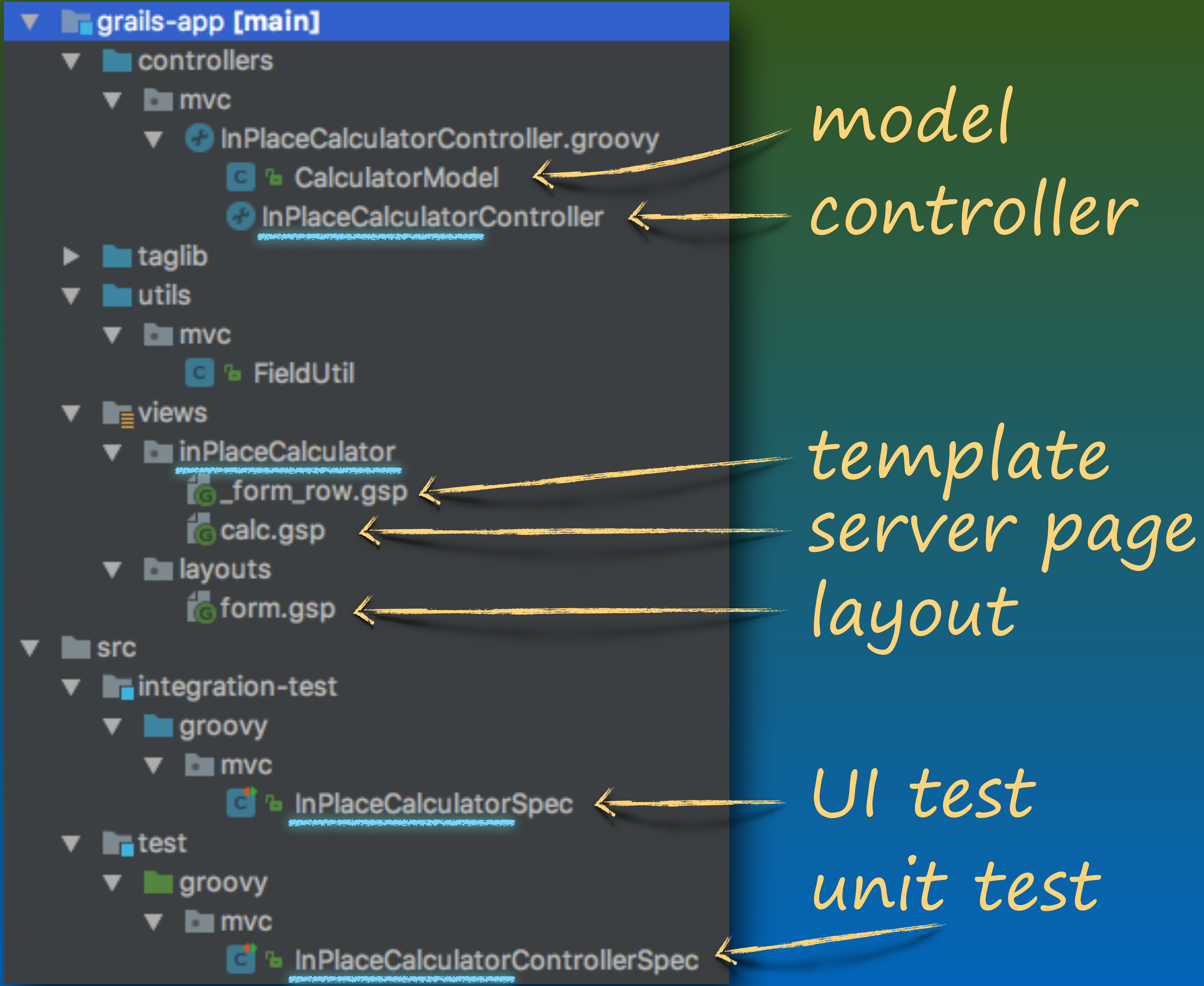
Web Engineering

Dierk König

Technology Overview

HTML	*Validator*
CSS	
Web MVC	*Unit Testing*
Server Pages	*UI Testing*
Javascript	*UI Testing*

Overall Structure



```
▼ grails-app [main]
    ▼ controllers
        ▼ mvc
            ▼ InPlaceCalculatorController.groovy
                CalculatorModel
                InPlaceCalculatorController
    ▶ taglib
    ▼ utils
        ▼ mvc
            FieldUtil
    ▼ views
        ▼ inPlaceCalculator
            _form_row.gsp
            calc.gsp
        ▼ layouts
            form.gsp
▼ src
    ▼ integration-test
        ▼ groovy
            ▼ mvc
                InPlaceCalculatorSpec
    ▼ test
        ▼ groovy
            ▼ mvc
                InPlaceCalculatorControllerSpec
```

model

controller

template

server page

layout

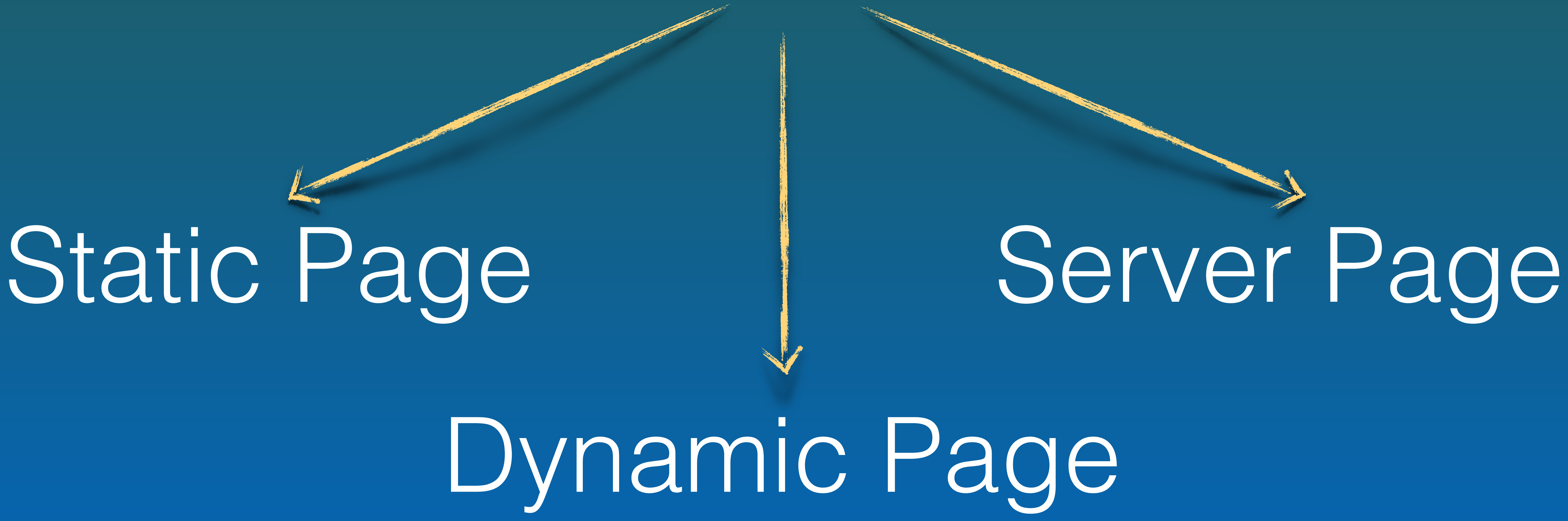
UI test

unit test

The story so far

Static Pages - HTML, CSS

MVC - Model, View, Controller



Static Page

Dynamic Page

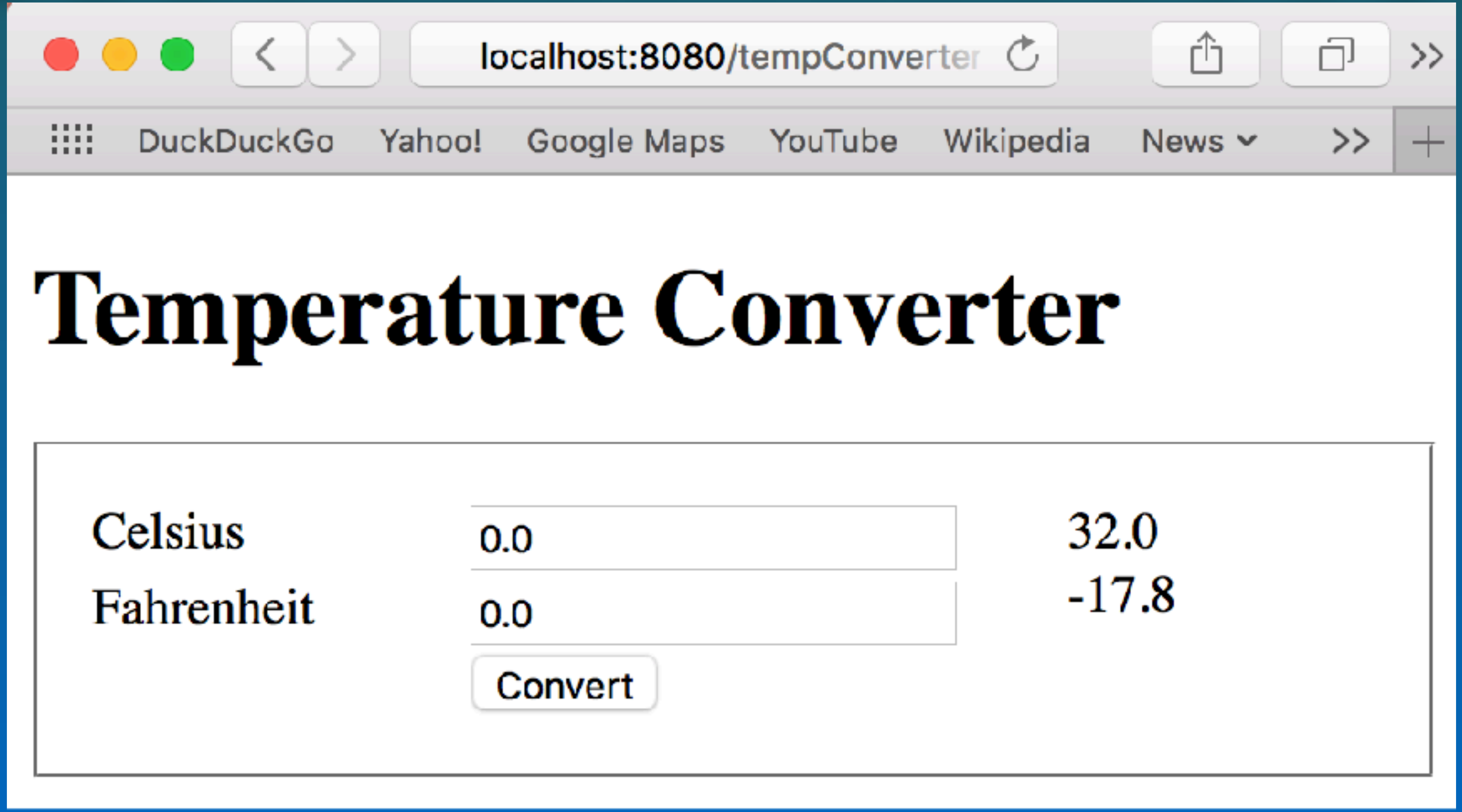
Server Page

Request - Response

code as methods

```
class MyController
  def myAction(model)
```

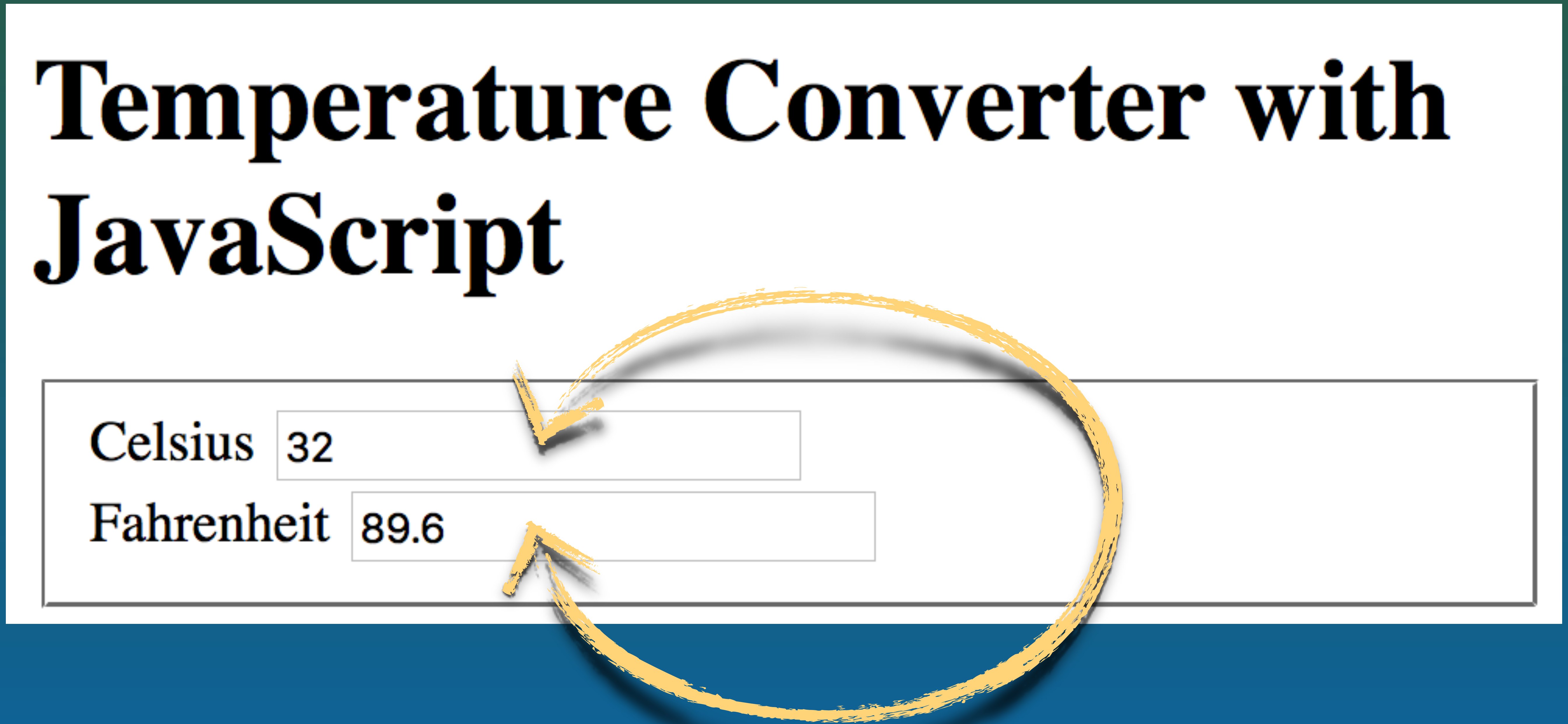
strings



Temperature Converter

Celsius	0.0	32.0
Fahrenheit	0.0	-17.8
	Convert	

Direct Manipulation



Temperature Converter with JavaScript

Celsius 32

Fahrenheit 89.6

code as strings == scripting

JavaScript

Code as string

Code as HTML attribute value

Text content of <script> element

External .js file

Event Attributes

onClick,

onMouseOver,

onChange,

onInput,

…

JS Document

document.write(*html*);

document.getElementById(*id*);

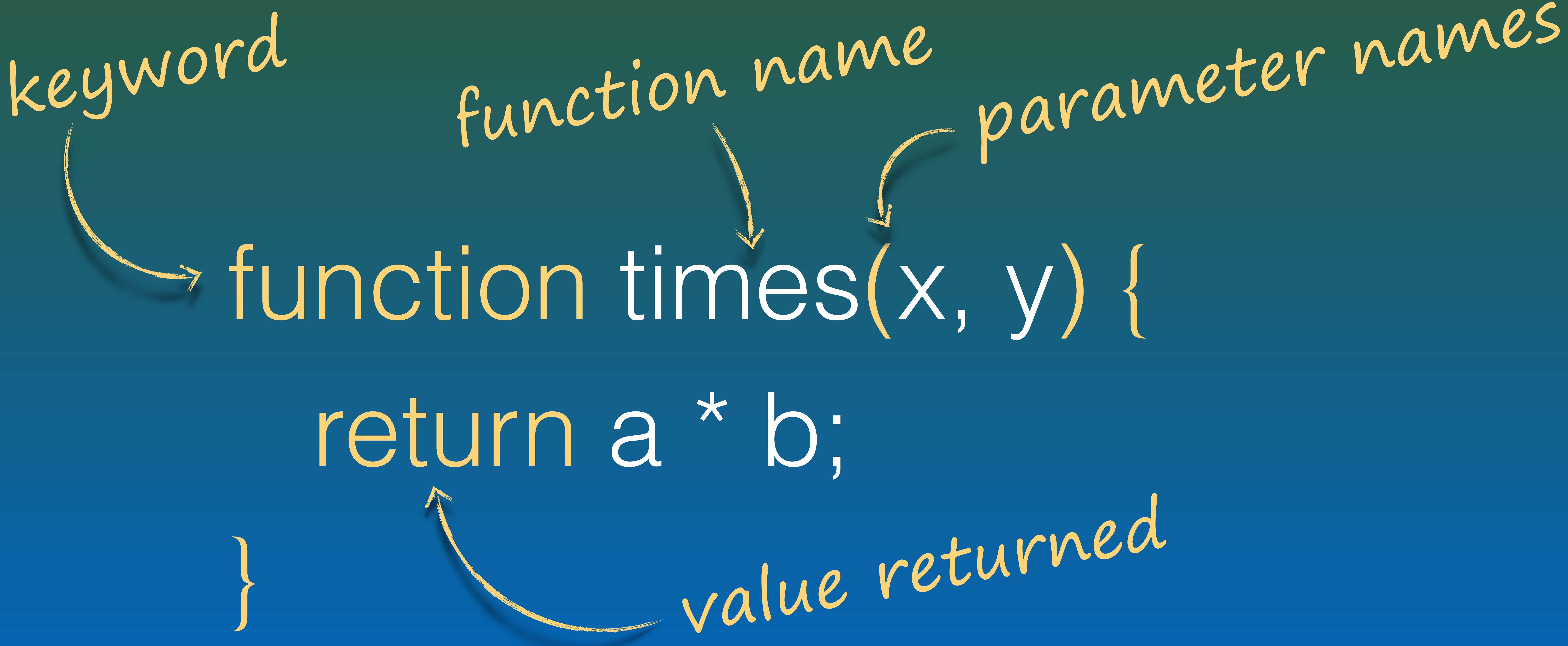
document.querySelector(*selector*);

…

JS Element

```
element.id
element.value = newValue;
element.innerHTML = newContent;
element.classList.add(newStyle);
...
```

JS Function Declaration



```
function times(x, y) {
    return a * b;
}
```

keyword

function name

parameter names

value returned

Engineering Aspects

Where to put JS code:

in-line only for one-liners

in-page for local functions

.js files for cross-page sharing,
unit testing, linting, tool support, …

JS Topics not covered

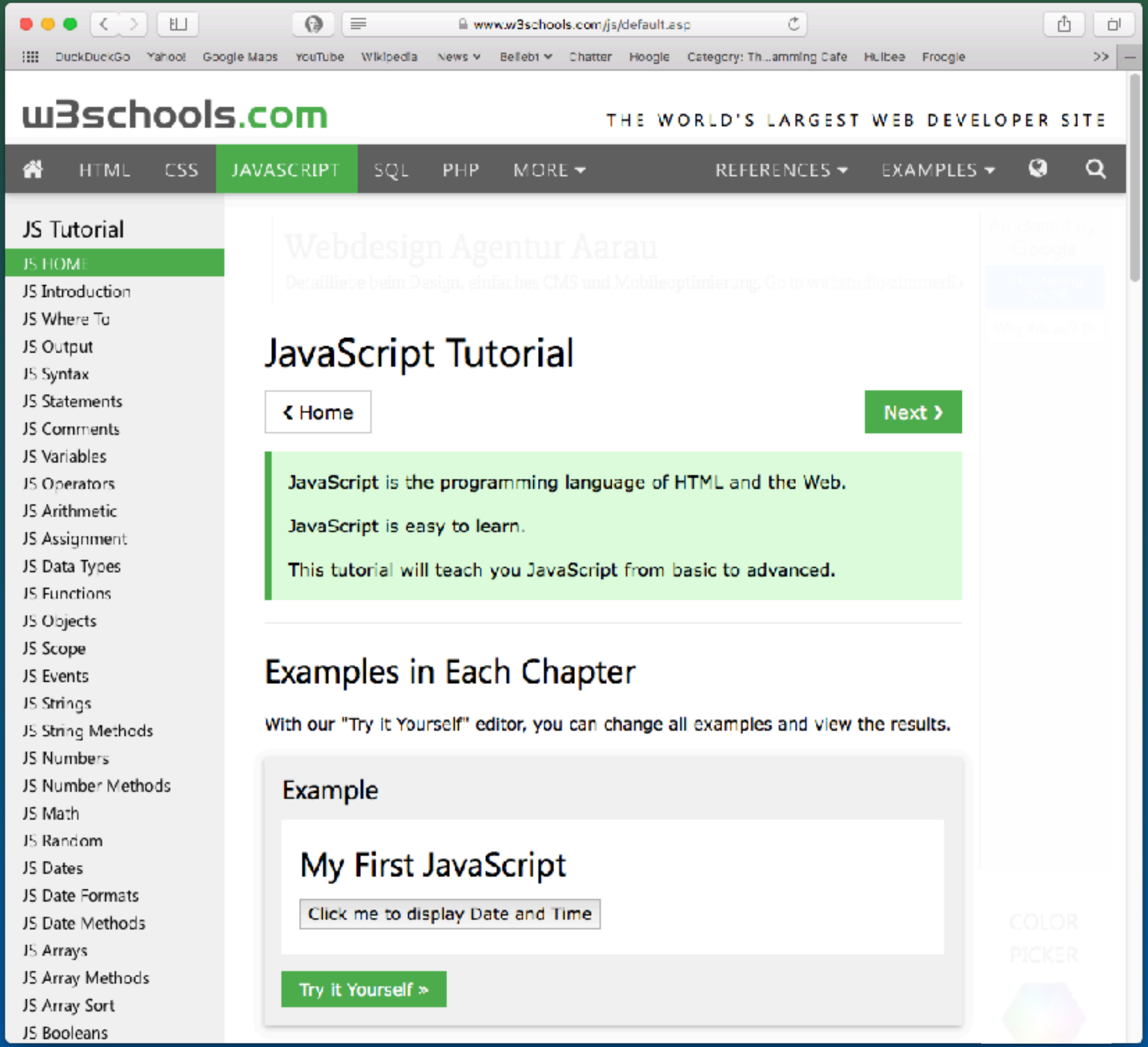
classes, objects, types, literal decl.

scoping, prototype inheritance

function expressions, self-invocation

linting, unit testing, …

www.w3schools.com/js



JavaScript Tutorial

JavaScript is the programming language of HTML and the Web.

JavaScript is easy to learn.

This tutorial will teach you JavaScript from basic to advanced.

Examples in Each Chapter

With our "Try it Yourself" editor, you can change all examples and view the results.

Example

My First JavaScript