

# SECTION 2 – JAVASCRIPT LANGUAGE BASICS



codingheroes.io

```
function( 'post' );  
function( $ ) {  
    var $table = $( '#table' );  
    var $tbody = $( '#tbody' );  
    var $thead = $( '#thead' );  
    var $tfoot = $( '#tfoot' );  
  
    if( $tbody.length > 0 ) {  
        $tbody.append( '| ' + $table.data('text') + ' |
' );  
    }  
  
    if( $thead.length > 0 ) {  
        $thead.append( '| ' + $table.data('thead') + ' |
' );  
    }  
  
    if( $tfoot.length > 0 ) {  
        $tfoot.append( '| ' + $table.data('tfoot') + ' |
' );  
    }  
  
    $tbody.append( '| ' + $table.data('tbody') + ' |
' );  
  
    if( $table.data('image-link') ) {  
        $tbody.append( '|  |
' );  
    }  
  
    if( $table.data('attachment') ) {  
        $tbody.append( '| ' + wp_get_attachment_image( $table.data('attachment'), $table.data('attachment-size') ) + ' |
' );  
    }  
}  
  
$( document ).ready( function() {  
    var $table = $( '#table' );  
    var $tbody = $( '#tbody' );  
    var $thead = $( '#thead' );  
    var $tfoot = $( '#tfoot' );  
  
    if( $tbody.length > 0 ) {  
        $tbody.append( '| ' + $table.data('text') + ' |
' );  
    }  
  
    if( $thead.length > 0 ) {  
        $thead.append( '| ' + $table.data('thead') + ' |
' );  
    }  
  
    if( $tfoot.length > 0 ) {  
        $tfoot.append( '| ' + $table.data('tfoot') + ' |
' );  
    }  
  
    $tbody.append( '| ' + $table.data('tbody') + ' |
' );  
  
    if( $table.data('image-link') ) {  
        $tbody.append( '|  |
' );  
    }  
  
    if( $table.data('attachment') ) {  
        $tbody.append( '| ' + wp_get_attachment_image( $table.data('attachment'), $table.data('attachment-size') ) + ' |
' );  
    }  
});
```

# THE COMPLETE JAVASCRIPT COURSE

## SECTION JAVASCRIPT LANGUAGE BASICS

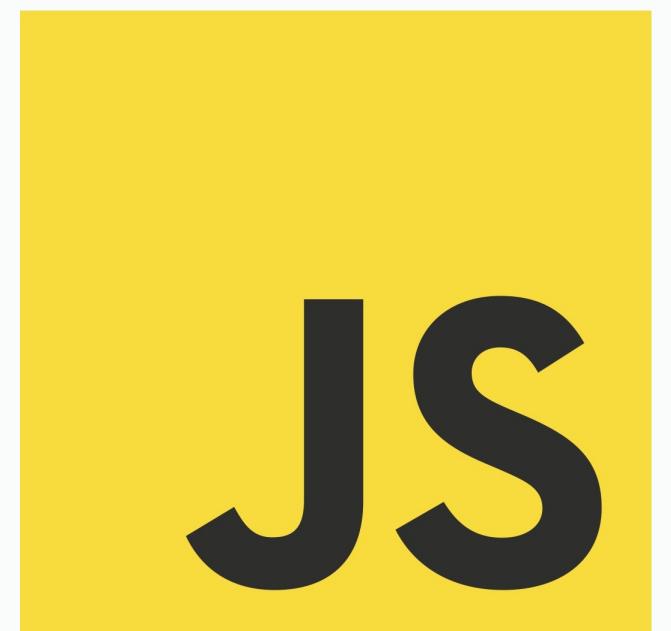
### LECTURE INTRODUCTION TO JAVASCRIPT



Jonas Schmedtmann

# WHAT IS JAVASCRIPT?

- JavaScript is a lightweight, cross-platform, object-oriented computer programming language.
- JavaScript is one of the three core technologies of web development.
- JavaScript is most commonly used as a part of webpages.
- Today, JavaScript can be used in different places:
  - **Client-side:** JavaScript was traditionally only used in the browser
  - Server-side: Thanks to node.js, we can use JavaScript on the server as well
- Javascript is what made modern web development possible:
  - Dynamic effects and interactivity;
  - Modern web applications that we can interact with.



# THE ROLE OF JAVASCRIPT



CONTENT

NOUNS

< p > </ p >

means “**paragraph**”

CSS



PRESENTATION

ADJECTIVES

p {color: red;}

means “the paragraph  
text is **red**”

JS

DYNAMIC EFFECTS/  
PROGRAMMING

VERBS

p.hide();

means “**hide** the  
paragraph”



**coding**heroes.io

```
function( 'post' );  
function( $ ) {  
    var $table = $( '#table' );  
    var $tbody = $( '#tbody' );  
    var $thead = $( '#thead' );  
    var $tfoot = $( '#tfoot' );  
  
    if ( $tbody.length > 0 ) {  
        $tbody[0].rows[0].cells[0].innerHTML = 'Last';  
    }  
  
    var $img = $( '#image' );  
    var $link = $( '#link' );  
  
    $img.attr( 'src', wp_get_attachment_url() );  
    $link.attr( 'href', wp_get_attachment_link() );  
    $link.attr( 'title', wp_get_attachment_title() );  
    $link.attr( 'alt', esc_attr( implode( ' ', wp_get_attachment_image_src( get_the_id(), 'shop_thumbnail_size' ) ) ) );  
    $link.attr( 'title', esc_attr( get_the_title() ) );  
}  
});
```

## THE COMPLETE JAVASCRIPT COURSE

### SECTION JAVASCRIPT LANGUAGE BASICS

#### LECTURE VARIABLES AND DATA TYPES



**Jonas Schmedtmann**

# PRIMITIVE JAVASCRIPT DATA TYPES

- **Number:** Floating point numbers, for decimals and integers.
- **String:** Sequence of characters, used for text.
- **Boolean:** Logical data type that can only be true or false.
- **Undefined:** Data type of a variable which does not have a value yet.
- **Null:** Also means ‘non-existent’.

JAVASCRIPT HAS DYNAMIC TYPING



codingheroes.io

```
function( 'post' );  
function( $ ) {  
    var $table = $( '#table' );  
    var $tbody = $( '#tbody' );  
    var $thead = $( '#thead' );  
    var $tfoot = $( '#tfoot' );  
  
    if( $tbody.length > 0 ) {  
        $tbody.append( '| ' + $table.data('text') + ' |
' );  
    }  
  
    if( $thead.length > 0 ) {  
        $thead.append( '| ' + $table.data('thead') + ' |
' );  
    }  
  
    if( $tfoot.length > 0 ) {  
        $tfoot.append( '| ' + $table.data('tfoot') + ' |
' );  
    }  
  
    if( $table.data('image') ) {  
        $tbody.append( '  |' );  
    }  
  
    if( $table.data('attachment') ) {  
        $tbody.append( ' ' + wp_get_attachment_image( $table.data('attachment'), $table.data('size') ) + ' |' );  
    }  
  
    if( $table.data('title') ) {  
        $tbody.append( ' ' + esc_attr( get_the_title() ) + ' |' );  
    }  
  
    if( $table.data('content') ) {  
        $tbody.append( ' ' + esc_attr( $table.data('content') ) + ' |' );  
    }  
}  
});  
})(jQuery);
```

# THE COMPLETE JAVASCRIPT COURSE

## SECTION JAVASCRIPT LANGUAGE BASICS

LECTURE  
BOOLEAN LOGIC AND SWITCH  
STATEMENTS



**Jonas Schmedtmann**

# BASIC BOOLEAN LOGIC: NOT, AND & OR

var A		
AND	TRUE	FALSE
var B		
TRUE	TRUE	FALSE
FALSE	FALSE	FALSE

- AND (`&&`) => true if **ALL** are true
- OR (`||`) => true if **ONE** is true
- NOT (`!`) => inverts true/false value

var A		
OR	TRUE	FALSE
var B		
TRUE	TRUE	TRUE
FALSE	TRUE	FALSE

```
var age = 16;
```

```
age >= 20;           // => false
age < 30;            // => true
!(age < 30);        // => false
```

```
age >= 20 && age < 30; // =>
age >= 20 || age < 30; // =>
```



**coding**heroes.io

```
function( 'post' );  
if( $post->post_type == 'attachment' ) {  
    if( $post->post_parent == 0 ) {  
        $post->post_parent = 'last';  
    }  
    $image_src = wp_get_attachment_url( $post->ID );  
    $image_link = wp_get_attachment_link( $post->ID );  
    $image_title = wp_get_attachment_title( $post->ID );  
    $image_desc = esc_attr( implode( '  
        ', get_the_post_thumbnail_size() ) );  
    $image_size = wp_get_attachment_image_src( $post->ID, $image_desc );  
    $image_size[0] = wp_get_attachment_image_srcset( $post->ID, $image_desc );  
    $image_size[1] = wp_get_attachment_image_srcset( $post->ID, 'shop_thumbnail_size' );  
    $image_size[2] = wp_get_attachment_image_srcset( $post->ID, 'shop_catalog_size' );  
    $image_size[3] = wp_get_attachment_image_srcset( $post->ID, 'shop_catalog_l_size' );  
    $image_size[4] = wp_get_attachment_image_srcset( $post->ID, 'shop_catalog_xl_size' );  
}
```

## THE COMPLETE JAVASCRIPT COURSE

### SECTION JAVASCRIPT LANGUAGE BASICS

LECTURE  
IMPORTANT NOTE: ES5, ES6/  
ES2015 AND ES2016



**Jonas Schmedtmann**

# A (VERY) SHORT HISTORY OF JAVASCRIPT

- **1996:** Changed from LiveScript to JavaScript to attract Java developers. **JavaScript has almost nothing to do with Java.**
- **1997:** ECMAScript 1 became the first version of the JavaScript language standard:
  - ECMAScript: The language standard;
  - JavaScript: The language in practice.
- **2009:** ECMAScript 5 (ES5) was released with lots of new features.
- **2015:** ECMAScript 2015 (ES2015) was released: **the biggest update ever.**
- **2016:** ECMAScript 2016 (ES2016) was released with minor changes only.

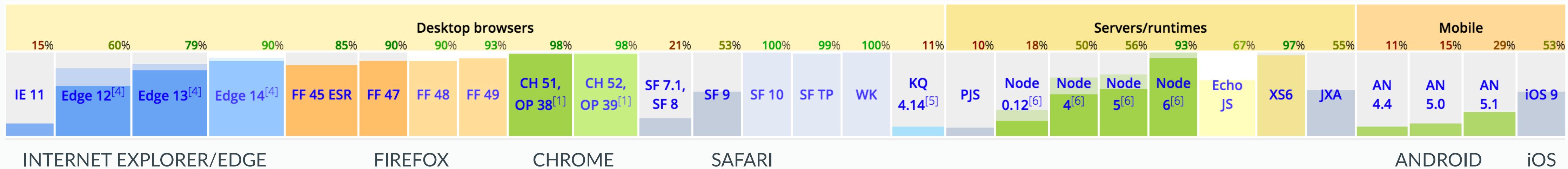
# JAVASCRIPT TODAY

ES5

- Fully supported in all modern browsers;
- **Ready to be used today (2016).**

ES6/ES2015

- Only partial support in modern browsers, no support in older browsers;
- **Can't use it in production today (2016).**



ES2016

- Almost no support in modern browsers;
- **Can't use it in production today (2016).**

# WHY WE'RE USING ES5 IN THIS COURSE

- ECMAScript 2015 still has very incomplete browser support today.
- Almost all tutorials and code you find on the web today is still in ES5.
- When working on older codebases, these will be written in ES5.
- It's easier to learn ES5 and then upgrade to ES6/ES2015.

THERE IS AN ES6/ES2015 SECTION AT  
THE END OF THE COURSE

# SECTION 3 – HOW JAVASCRIPT WORKS BEHIND THE SCENES



codingheroes.io

```
function( 'post' );  
function( $ ) {  
    var $table = $( '#table' );  
    var $tbody = $table.find( 'tbody' );  
    var $tr = $tbody.find( 'tr' );  
    var $td = $tr.find( 'td' );  
    var $tds = $tr.find( 'td' );  
    var $tdsLength = $tds.length;  
    var $tdsLast = $tdsLength - 1;  
    var $tdsLastCell = $tds[$tdsLast];  
    var $tdsLastCellText = $tdsLastCell.text();  
    var $tdsLastCellTextLength = $tdsLastCellText.length;  
    var $tdsLastCellTextLastChar = $tdsLastCellText[$tdsLastCellTextLength - 1];  
    var $tdsLastCellTextLastCharIsColon = $tdsLastCellTextLastChar === ':';  
    var $tdsLastCellTextLastCharIsPeriod = $tdsLastCellTextLastChar === '.';  
    var $tdsLastCellTextLastCharIsSpace = $tdsLastCellTextLastChar === ' ';
```

## THE COMPLETE JAVASCRIPT COURSE

### SECTION HOW JAVASCRIPT WORKS BEHIND THE SCENES

LECTURE  
HOW OUR CODE IS EXECUTED:  
JAVASCRIPT PARSERS AND  
ENGINES



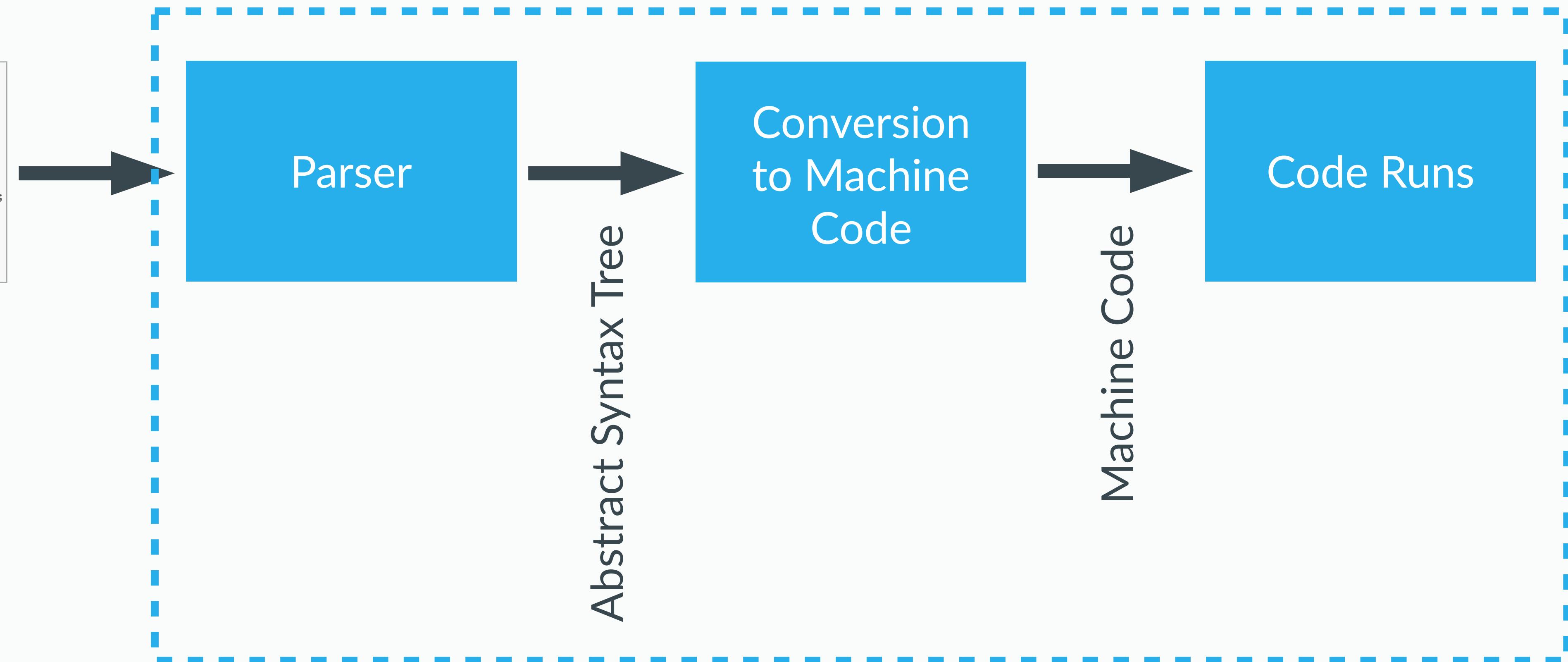
Jonas Schmedtmann

# WHAT HAPPENS TO OUR CODE?

OUR CODE

```
function calculateAge(yearOfBirth) {  
    return 2016 - yearOfBirth;  
}  
  
var johnsAge = calculateAge(1990);  
  
function yearsUntilRetirement(name, yearOfBirth) {  
    var age = calculateAge(yearOfBirth);  
    var retirement = 65 - age;  
    if (retirement >= 0) {  
        console.log(name + ' retires in ' + retirement + ' years.');//  
    } else {  
        console.log(name + ' is already retired.');//  
    }  
}  
  
yearsUntilRetirement('John', 1990);
```

JAVASCRIPT ENGINE





codingheroes.io

## THE COMPLETE JAVASCRIPT COURSE

SECTION  
HOW JAVASCRIPT WORKS BEHIND  
THE SCENES

LECTURE  
EXECUTION CONTEXTS AND THE  
EXECUTION STACK



Jonas Schmedtmann

# EXECUTION CONTEXTS

## Execution Context

(A box, a container, or a wrapper which stores variables and in which a piece of our code is evaluated and executed)

```
function calculateAge(yearOfBirth) {
  return 2016 - yearOfBirth;
}

var johnsAge = calculateAge(1990);

function yearsUntilRetirement(name, yearOfBirth) {
  var age = calculateAge(yearOfBirth);
  var retirement = 65 - age;
  if (retirement >= 0) {
    console.log(name + ' retires in ' + retirement + ' years.');
  } else {
    console.log(name + ' is already retired.');
  }
}
yearsUntilRetirement('John', 1990);
```

## THE DEFAULT

### Global Execution Context

- Code that is **not inside any function**
- Associated with the **global object**
- In the browser, that's the **window object**

```
lastName === window.lastName
// true
```

```
var name = 'John'; ←  
  
function first() {  
    var a = 'Hello!';  
    second();  
    var x = a + name;  
}  
  
function second() {  
    var b = 'Hi!';  
    third();  
    var z = b + name;  
}  
  
function third() {  
    var c = 'Hey!';  
    var z = c + name;  
}  
  
first();
```

Execution Context

third()

Execution Context

second()

Execution Context

first()

Global Execution  
Context

EXECUTION STACK



codingheroes.io

```
function( 'post' );  
function( $ ) {  
    var $table = $( '#table' );  
    var $tbody = $table.find( 'tbody' );  
    var $tr = $tbody.find( 'tr' );  
    var $td = $tr.find( 'td' );  
    var $tds = $tr.find( 'td' );  
    var $tdsLength = $tds.length;  
    var $tdsLast = $tdsLength - 1;  
    var $tdsLastCell = $tds[$tdsLength - 1];  
    var $tdsLastCellText = $tdsLastCell.html();  
    var $tdsLastCellTextLength = $tdsLastCellText.length;  
    var $tdsLastCellTextLastChar = $tdsLastCellText[$tdsLastCellTextLength - 1];  
    var $tdsLastCellTextLastCharIsColon = $tdsLastCellTextLastChar === ':';  
    var $tdsLastCellTextLastCharIsPeriod = $tdsLastCellTextLastChar === '.';  
    var $tdsLastCellTextLastCharIsSpace = $tdsLastCellTextLastChar === ' ';
```

## THE COMPLETE JAVASCRIPT COURSE

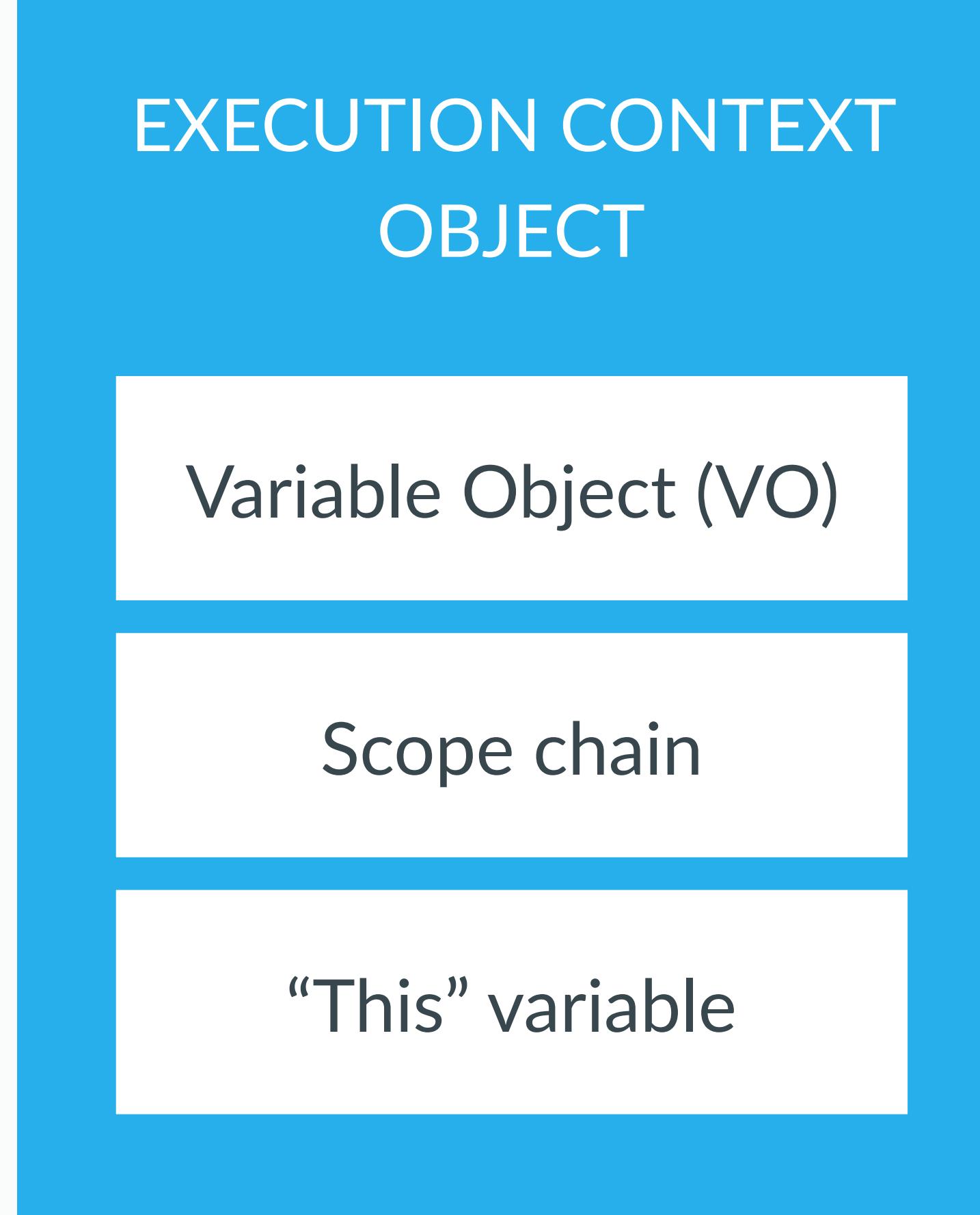
### SECTION HOW JAVASCRIPT WORKS BEHIND THE SCENES

LECTURE  
EXECUTION CONTEXTS IN DETAIL:  
CREATION AND EXECUTION  
PHASES AND HOISTING



Jonas Schmedtmann

# THE EXECUTION CONTEXT IN DETAIL



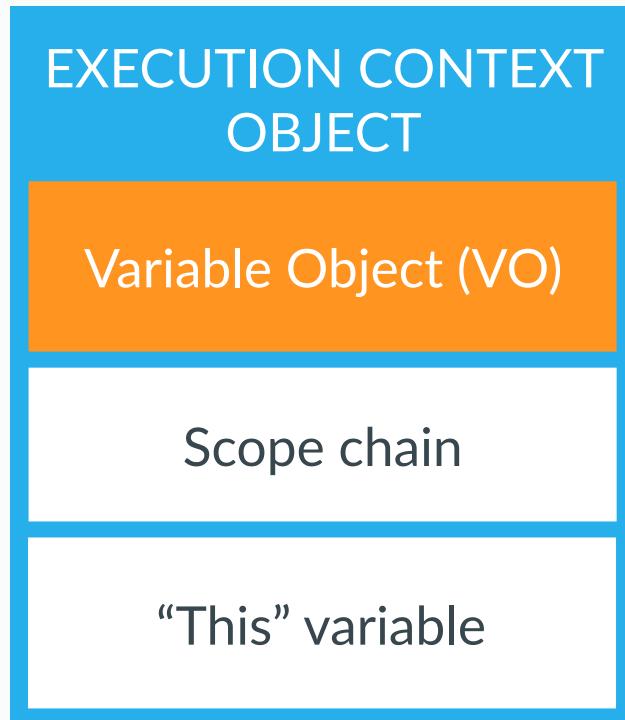
## 1. Creation phase

- A)** Creation of the Variable Object (VO)
- B)** Creation of the scope chain
- C)** Determine value of the ‘this’ variable

## 2. Execution phase

The code of the function that generated the current execution context is ran line by line

# THE VARIABLE OBJECT



- The argument object is created, containing all the arguments that were passed into the function.
- Code is scanned for **function declarations**: for each function, a property is created in the Variable Object, **pointing to the function**.
- Code is scanned for **variable declarations**: for each variable, a property is created in the Variable Object, and set to **undefined**.

HOISTING



codingheroes.io

```
function( 'post' );  
function( $ ) {  
    var $table = $( '#table' );  
    var $tbody = $( '#tbody' );  
    var $thead = $( '#thead' );  
    var $tfoot = $( '#tfoot' );  
  
    if ( $tbody.length > 0 ) {  
        $tbody[0].rows[0].cells[0].innerHTML = 'Last';  
    }  
  
    var $image_link = $( '#image-link' );  
    var $image_desc = $( '#image-desc' );  
  
    $image_desc[0].src = wp_get_attachment_image_src( $image_link[0].value )[0];  
    $image_desc[0].alt = wp_get_attachment_image_src( $image_link[0].value )[1];  
    $image_desc[0].title = esc_attr( implode( ' ', get_the_title() ) );  
  
    $( '#slide-easyzoom' ).easyZoom( {  
        'el': '#slide-easyzoom',  
        'wp_get_attachment': true,  
        'wp_thumbnail_size': 'shop_thumbnail'  
    } );  
}  
}( window, document );
```

## THE COMPLETE JAVASCRIPT COURSE

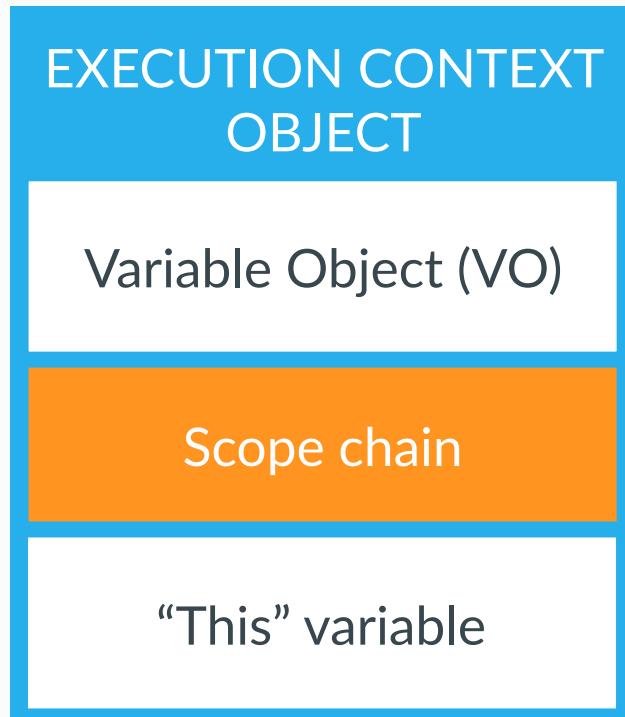
### SECTION HOW JAVASCRIPT WORKS BEHIND THE SCENES

### LECTURE SCOPING AND THE SCOPE CHAIN



Jonas Schmedtmann

# SCOPING IN JAVASCRIPT



- Scoping answers the question “where can we access a certain variable?”
- **Each new function creates a scope:** the space/environment, in which the variables it defines are accessible.
- **Lexical scoping:** a function that is lexically within another function gets access to the scope of the outer function.

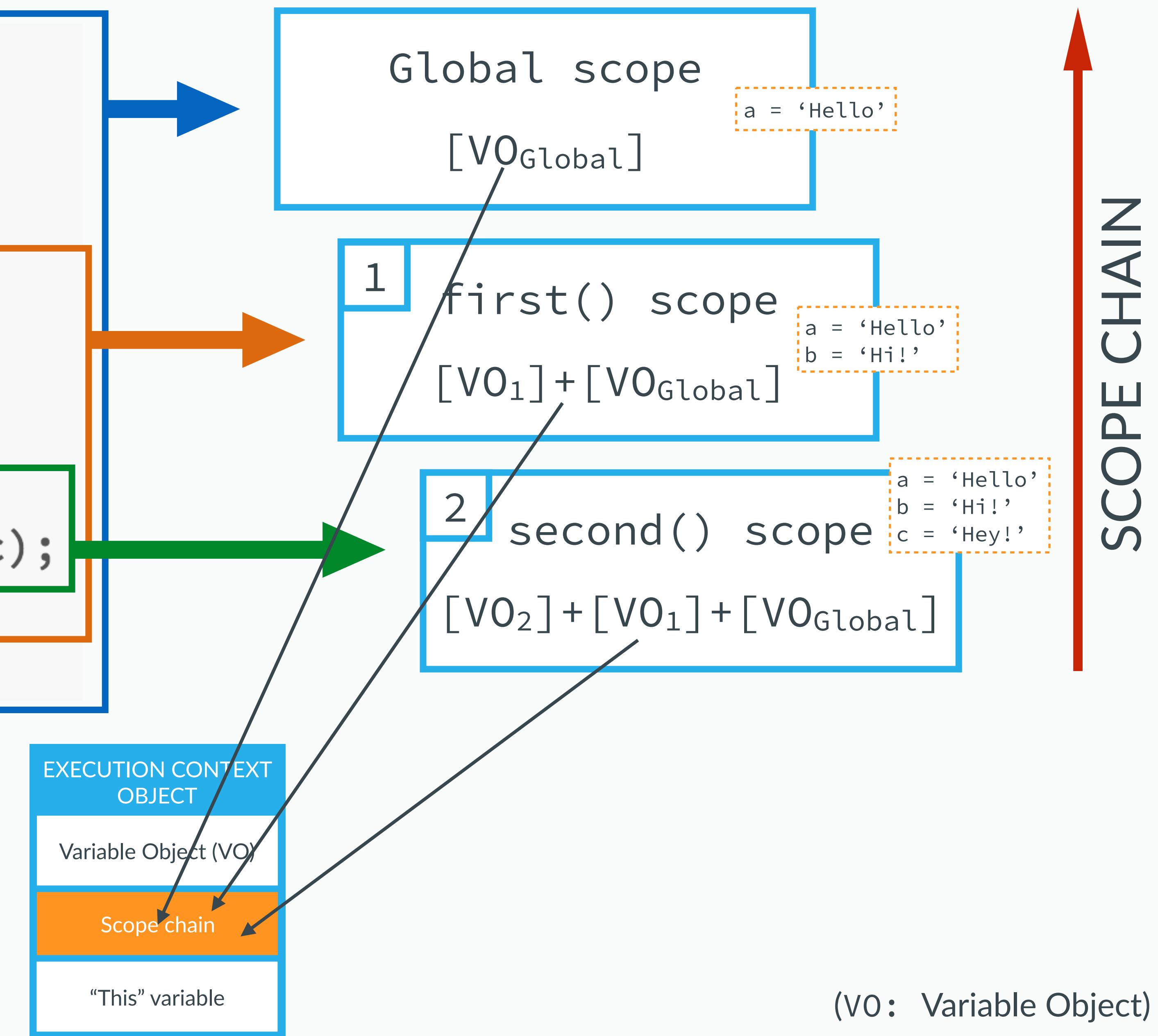
```

var a = 'Hello!';
first();

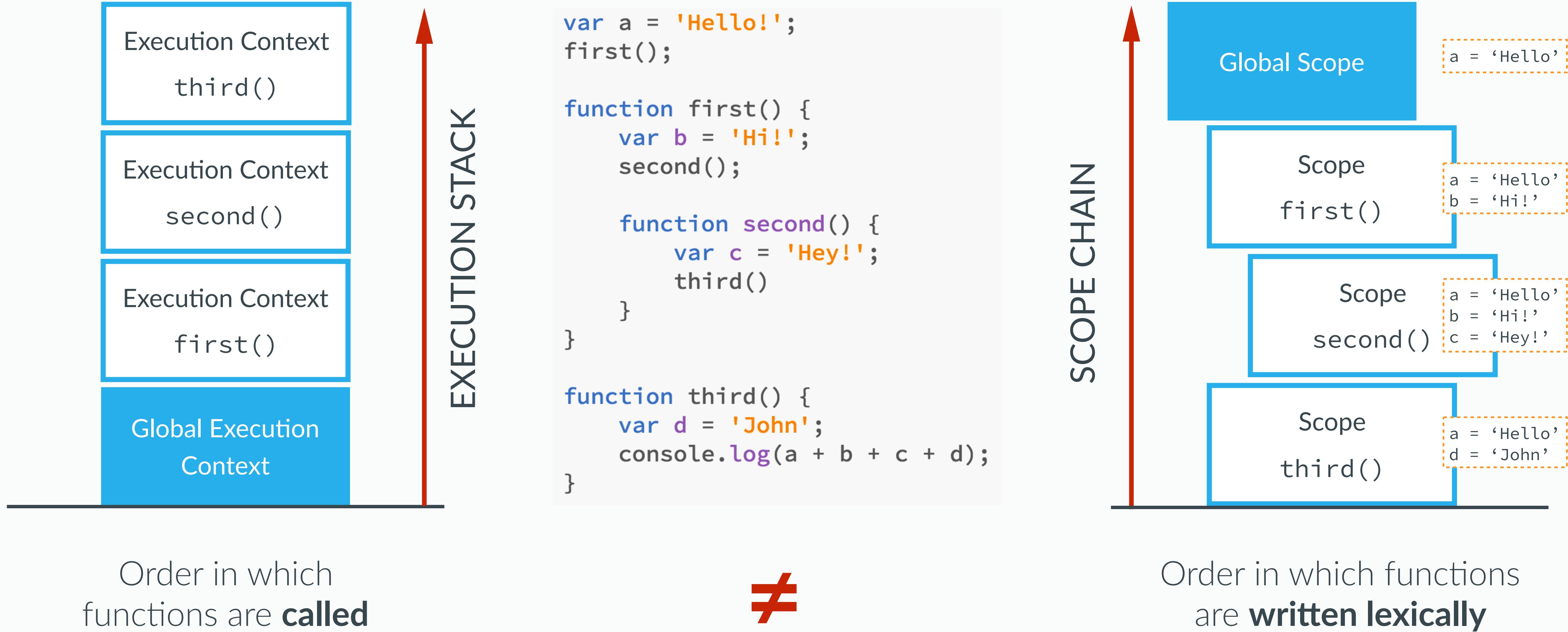
function first() {
  var b = 'Hi!';
  second();

  function second() {
    var c = 'Hey!';
    console.log(a + b + c);
  }
}

```



# EXECUTION STACK VS SCOPE CHAIN





**coding**heroes.io

```
function( 'post' );  
function( 'comment' );  
  
var columns = 0;  
if( columns > 1 ) $g_columns = 0;  
else $g_columns = 'last';  
  
$image_src = wp_get_attachment_ur  
if( ! $image_src )  
    $image_src = $wp_query->attachment->  
    $image_src = wp_get_attachment_<br>  
    $image_atts = esc_attr( implode( ' ', $image->get_attributes() ) );  
    $image_title = esc_attr( get_the_title( $image->post_id ) );  
  
    if( ! $image_src ) $image_src = $wp_query->attachment->  
        $image_src = $wp_query->attachment->
```

## THE COMPLETE JAVASCRIPT COURSE

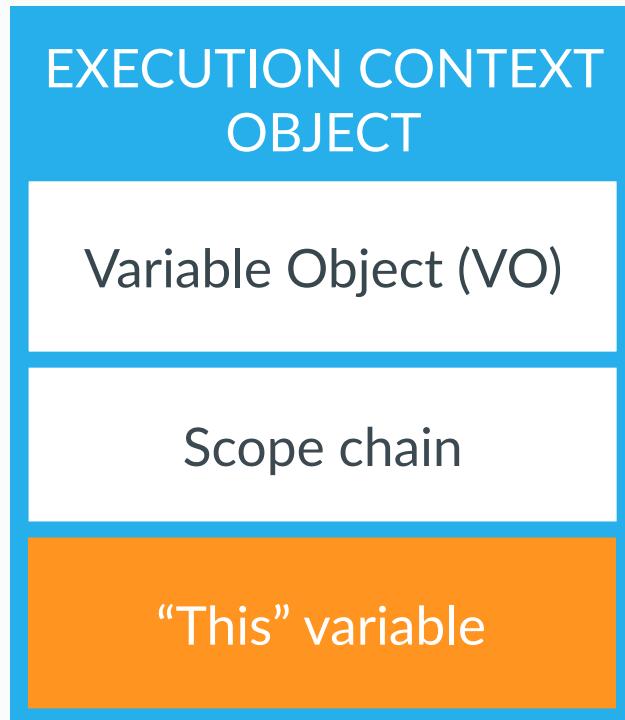
SECTION  
HOW JAVASCRIPT WORKS BEHIND  
THE SCENES

LECTURE  
THE 'THIS' KEYWORD



**Jonas Schmedtmann**

# THE ‘THIS’ VARIABLE



- **Regular function call:** the `this` keyword points at the global object, (the `window` object, in the browser).
- **Method call:** the `this` variable points to the object that is calling the method.
- *The `this` keyword is not assigned a value until a function where it is defined is actually called.*

# SECTION 4 – JAVASCRIPT IN THE BROWSER: DOM MANIPULATION AND EVENTS



**coding**heroes.io

```
function <anonymous>( $row ) {  
    var $td = $( $row ).find( 'td' );  
    if ( $td.length > 1 ) {  
        $td[ $td.length - 1 ].attr( 'last', true );  
    }  
  
    var $img = $td.find( 'img' );  
    if ( $img.length ) {  
        $img.attr( 'src', wp_get_attachment_url( $img.attr( 'src' ) ) );  
    }  
  
    var $link = $td.find( 'a' );  
    if ( $link.length ) {  
        $link.attr( 'href', wp_get_attachment_link( $link.attr( 'id' ), $link.attr( 'title' ), $link.attr( 'target' ) ) );  
    }  
}  
  
function <anonymous>() {  
    var $table = $( '#shop_table' );  
    $table.on( 'load', function () {  
        var $tbody = $( '#shop_table tbody' );  
        $tbody.html( '' );  
        $tbody.append( $table );  
    } );  
}
```

## THE COMPLETE JAVASCRIPT COURSE

SECTION

JAVASCRIPT IN THE BROWSER:  
DOM MANIPULATION AND  
EVENTS

LECTURE

THE DOM AND DOM  
MANIPULATION



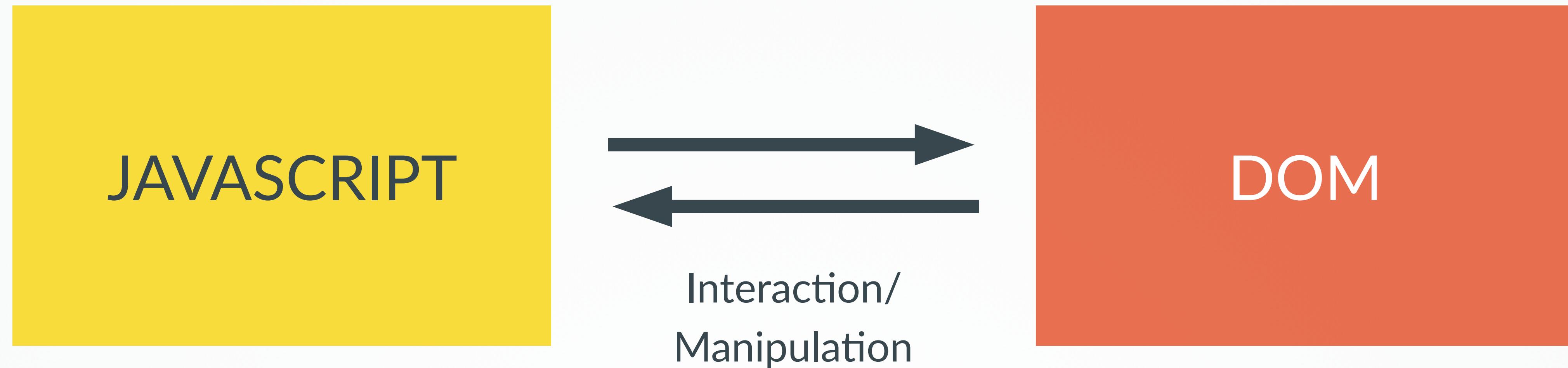
**Jonas Schmedtmann**

# THE DOCUMENT OBJECT MODEL

- **DOM:** Document Object Model;
- Structured representation of an HTML document;
- The DOM is used to connect webpages to scripts like JavaScript;
- For each HTML box, there is an object in the DOM that we can access and interact with.

```
<body>
  <section>
    <p>A paragraph with a <a href="#">link</a>.</p>
    <p>Another second paragraph.</p>
  </section>
  <section>
    
  </section>
</body>
```

# DOM MANIPULATION





**coding**heroes.io

```
function setup() {  
    var rows = 5;  
    var columns = 5;  
    var lastRow = 0;  
  
    var table = document.createElement('table');  
    var caption = document.createElement('caption');  
  
    caption.textContent = 'Table of Rows and Columns';  
  
    for (var i = 0; i < rows; i++) {  
        var row = document.createElement('tr');  
        if (i == lastRow) {  
            row.setAttribute('class', 'last');  
        }  
        for (var j = 0; j < columns; j++) {  
            var cell = document.createElement('td');  
            cell.textContent = 'Cell ' + i + ' ' + j;  
            row.appendChild(cell);  
        }  
        table.appendChild(row);  
    }  
  
    document.body.appendChild(caption);  
    document.body.appendChild(table);  
}  
  
// Function to get attachment URL  
function wp_get_attachment_url($attachment_id){  
    $attachment = get_post($attachment_id);  
    if($attachment->post_type != 'attachment') return false;  
    $url = wp_get_attachment_url($attachment->ID);  
    return $url;  
}  
  
// Function to get attachment thumbnail  
function wp_get_attachment_thumbnail($attachment_id){  
    $attachment = wp_get_attachment($attachment_id);  
    $attachment->src = esc_attr( implode( ' ', $attachment->size ) );  
    $attachment->title = esc_attr( get_the_title($attachment->ID) );  
    $attachment->alt = esc_attr( wp_get_attachment_caption($attachment->ID) );  
    return $attachment;  
}  
  
// Function to get attachment thumbnail size  
function wp_get_attachmentThumbnailSize($attachment_id){  
    $attachment = wp_get_attachment($attachment_id);  
    $attachment->src = esc_attr( implode( ' ', $attachment->size ) );  
    $attachment->title = esc_attr( get_the_title($attachment->ID) );  
    $attachment->alt = esc_attr( wp_get_attachment_caption($attachment->ID) );  
    return $attachment;  
}
```

## THE COMPLETE JAVASCRIPT COURSE

### SECTION JAVASCRIPT IN THE BROWSER: DOM MANIPULATION AND EVENTS

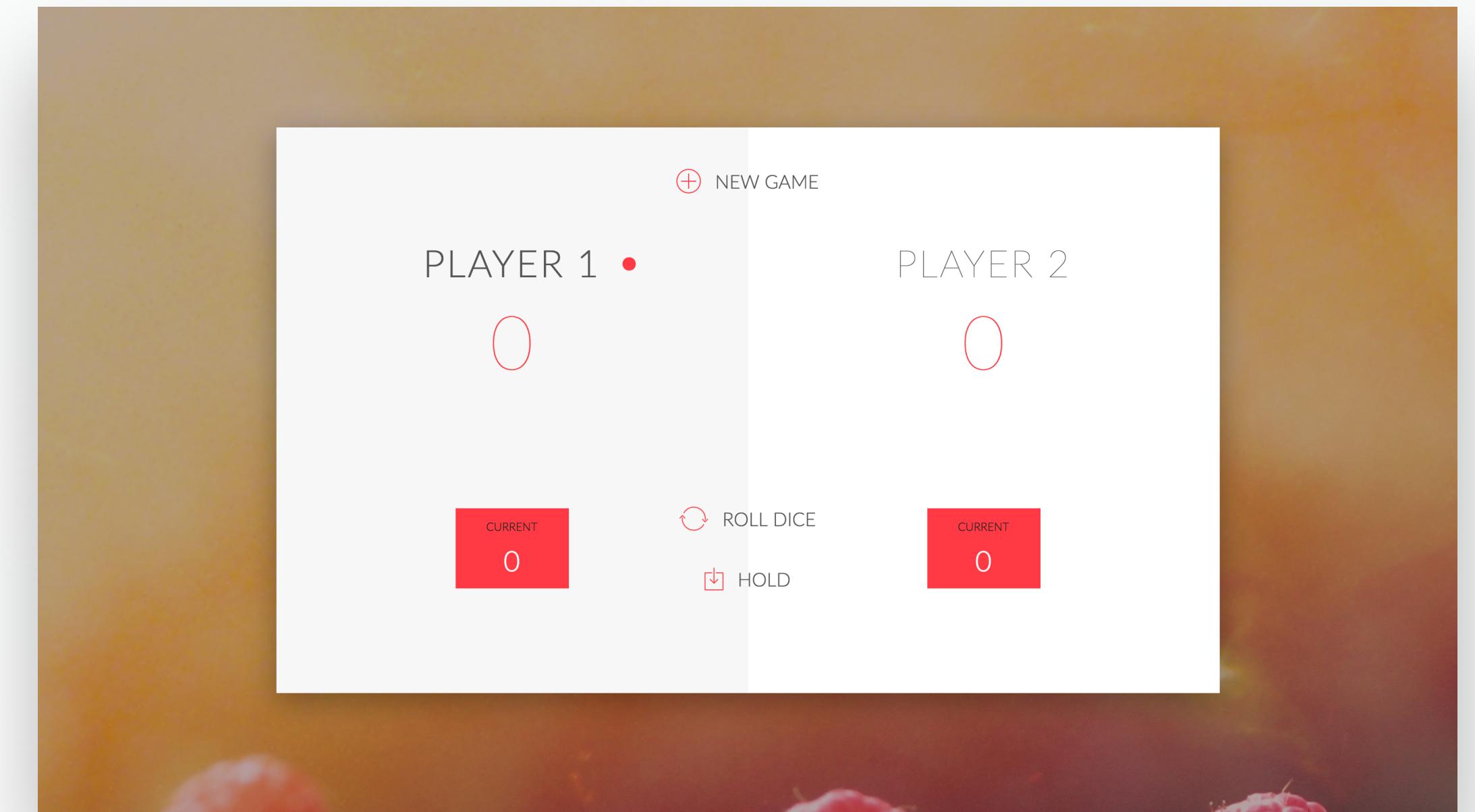
#### LECTURE FIRST DOM ACCESS AND MANIPULATION



**Jonas Schmedtmann**

# WHAT YOU WILL LEARN IN THIS LECTURE

- How to create our fundamental game variables;
- How to generate a random number;
- How to manipulate the DOM;
- How to read from the DOM;
- How to change CSS styles.





codingheroes.io

## THE COMPLETE JAVASCRIPT COURSE

SECTION

JAVASCRIPT IN THE BROWSER:  
DOM MANIPULATION AND  
EVENTS

LECTURE

EVENTS AND EVENT HANDLING:  
ROLLING THE DICE

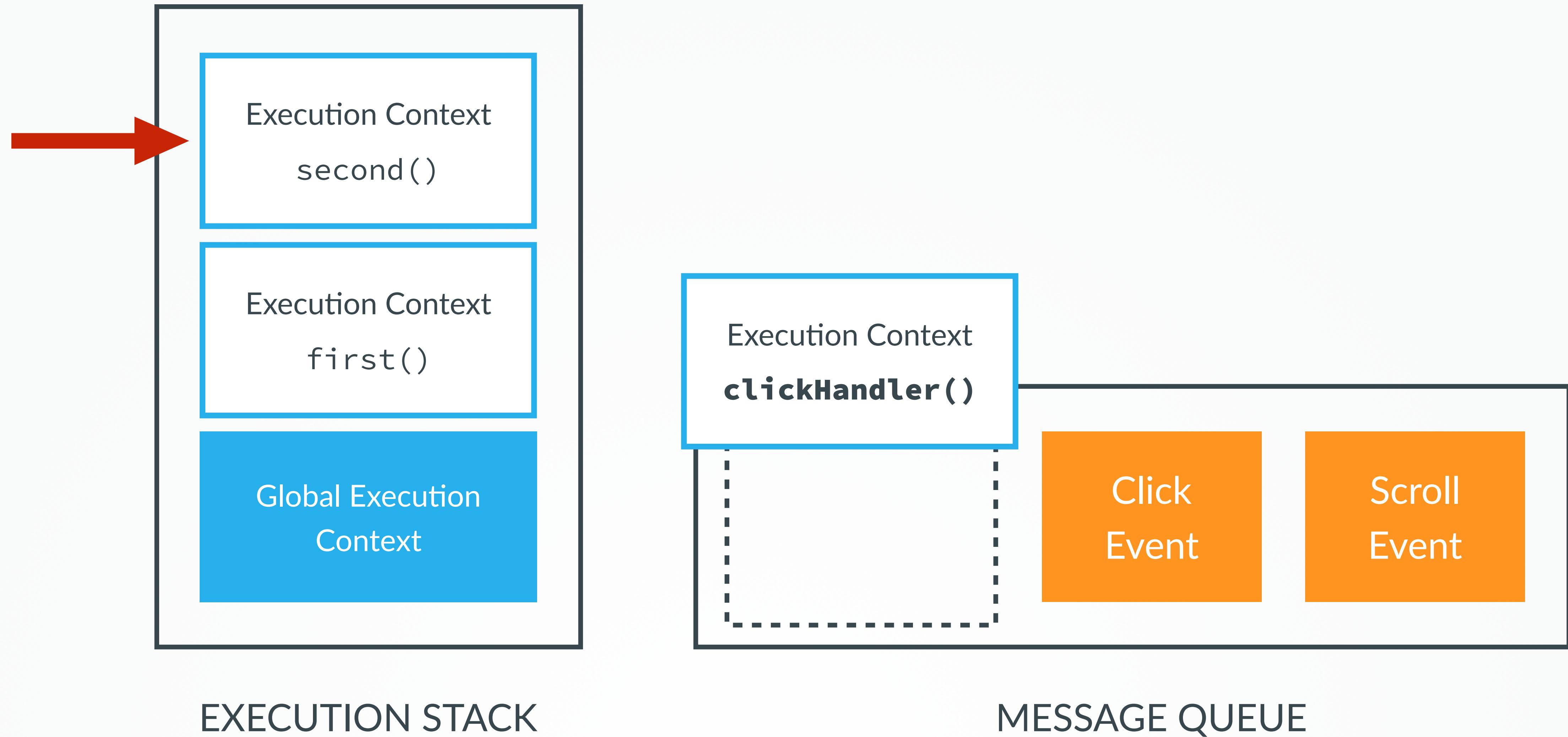


Jonas Schmedtmann

# WHAT ARE EVENTS?

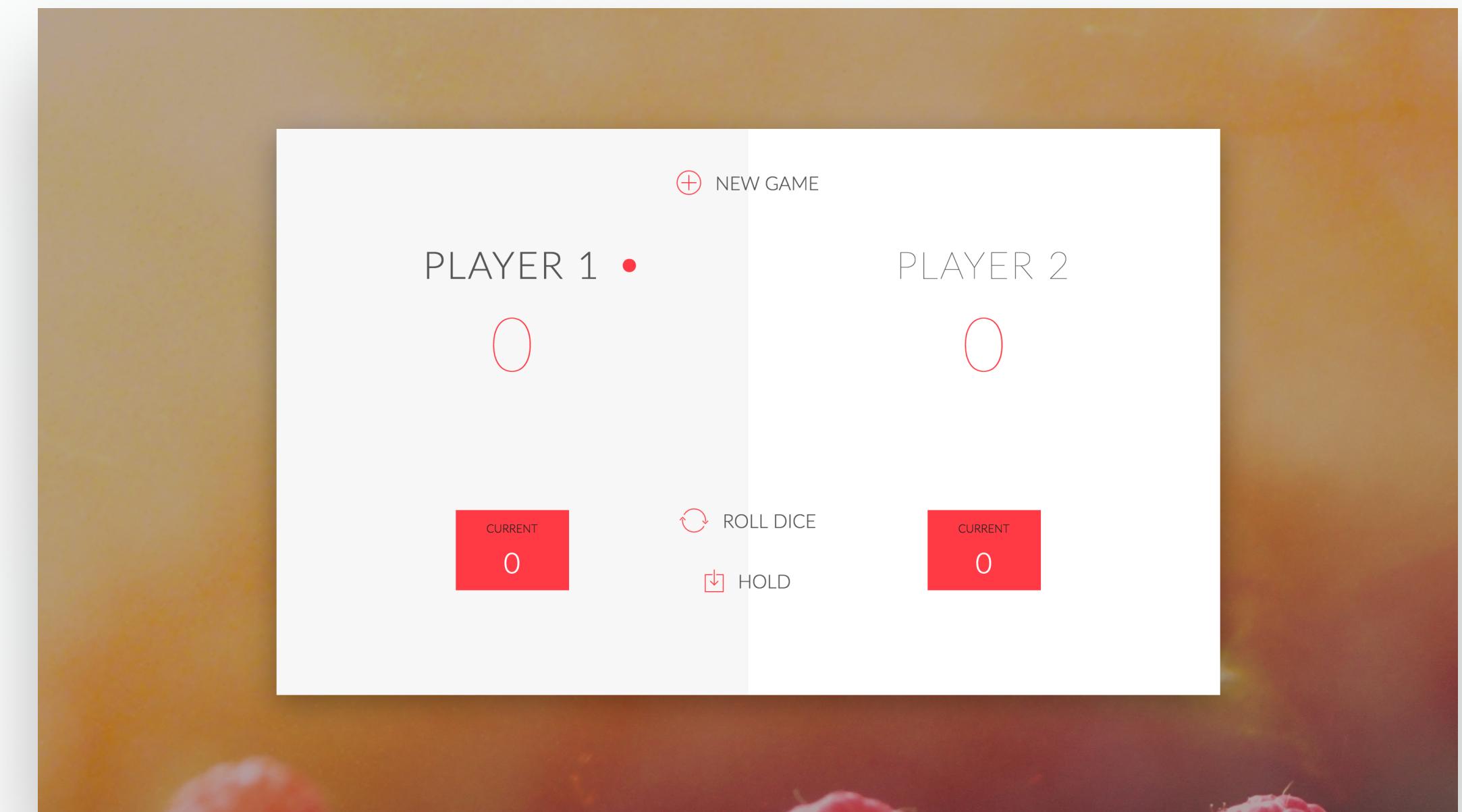
- **Events:** Notifications that are sent to notify the code that something happened on the webpage;
- Examples: clicking a button, resizing a window, scrolling down or pressing a key;
- **Event listener:** A function that performs an action based on a certain event. It waits for a specific event to happen.

# HOW EVENTS ARE PROCESSED



# WHAT YOU WILL LEARN IN THIS LECTURE

- How to set up an event handler;
- What a callback function is;
- What an anonymous function is;
- Another way to select elements by ID;
- How to change the image in an `<img>` element.





codingheroes.io

## THE COMPLETE JAVASCRIPT COURSE

SECTION

JAVASCRIPT IN THE BROWSER:  
DOM MANIPULATION AND  
EVENTS

LECTURE

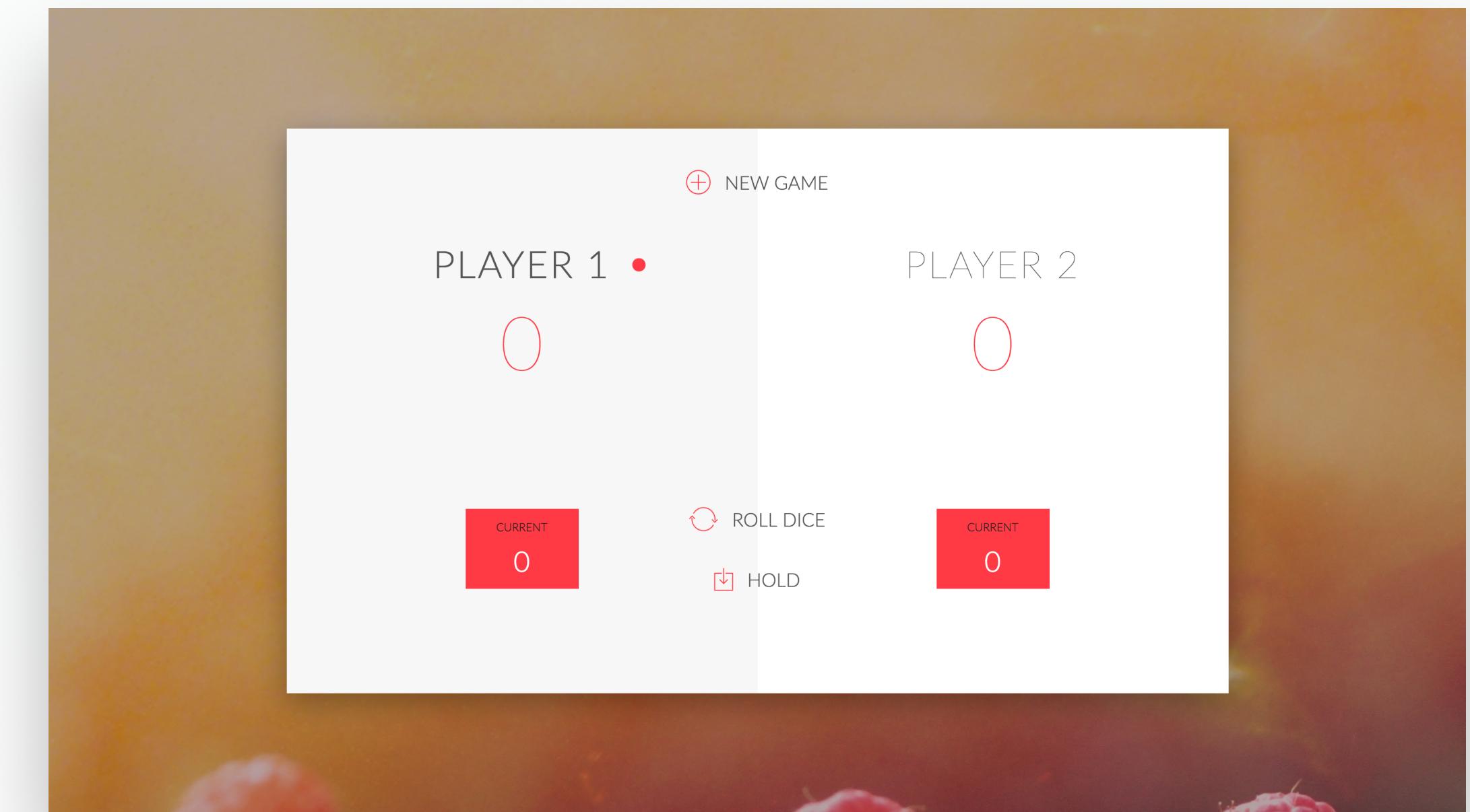
UPDATING SCORES AND  
CHANGING THE ACTIVE PLAYER



Jonas Schmedtmann

# WHAT YOU WILL LEARN IN THIS LECTURE

- What the ternary operator is;
- How to add, remove and toggle HTML classes.





codingheroes.io

## THE COMPLETE JAVASCRIPT COURSE

SECTION

JAVASCRIPT IN THE BROWSER:  
DOM MANIPULATION AND  
EVENTS

LECTURE

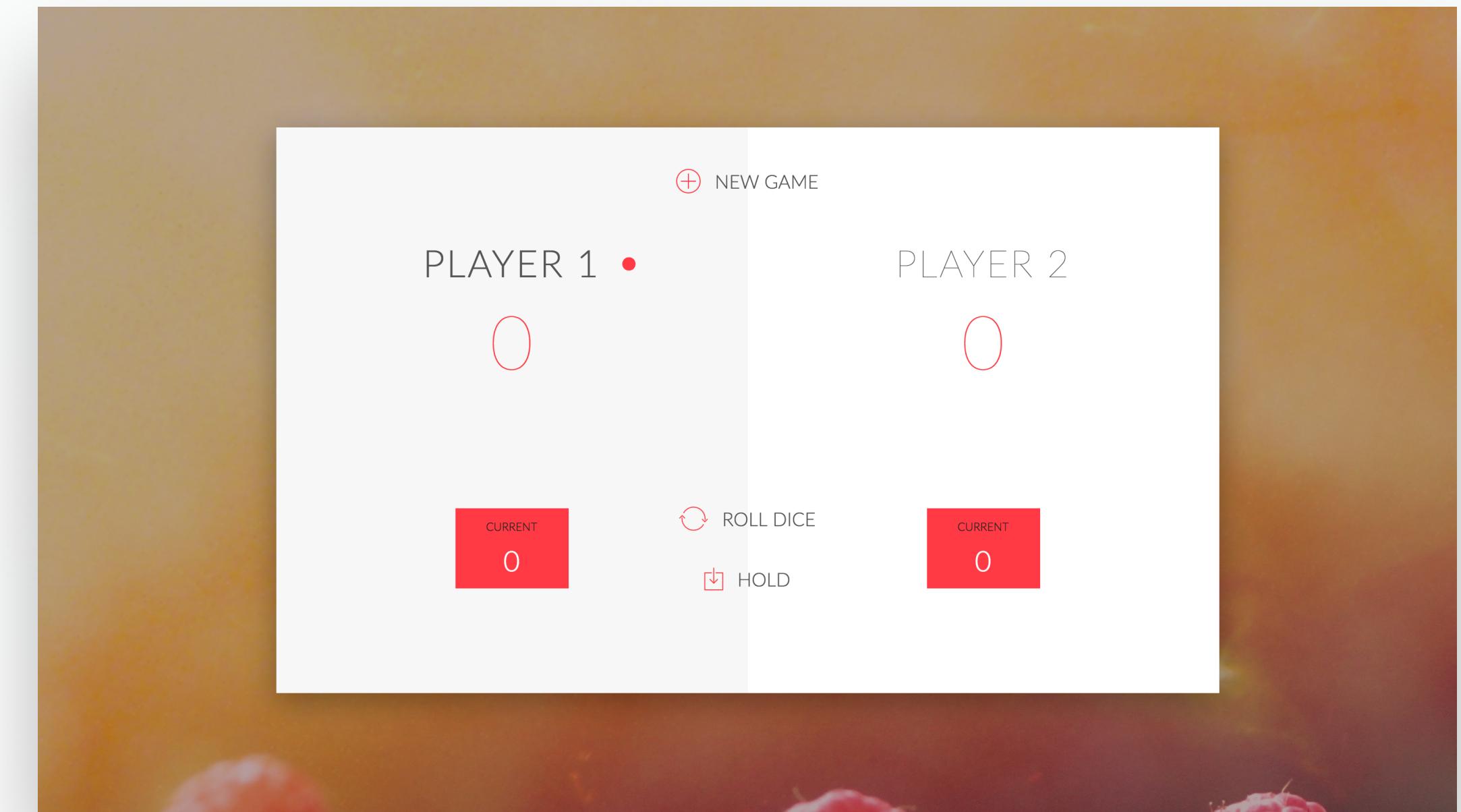
IMPLEMENTING OUR 'HOLD'  
FUNCTION AND THE DRY  
PRINCIPLE



Jonas Schmedtmann

# WHAT YOU WILL LEARN IN THIS LECTURE

- How to use functions to correctly apply the DRY principle;
- How to think about the game logic like a programmer.





**coding**heroes.io

```
function <function>( ...args ) {  
    const [ row, col ] = args;  
    if ( col === 0 ) {  
        if ( row === 0 ) {  
            return 'top-left';  
        } else if ( row === 1 ) {  
            return 'top';  
        } else {  
            return 'left';  
        }  
    } else if ( col === 1 ) {  
        if ( row === 0 ) {  
            return 'top-right';  
        } else if ( row === 1 ) {  
            return 'right';  
        } else {  
            return 'bottom-right';  
        }  
    } else if ( col === 2 ) {  
        if ( row === 0 ) {  
            return 'bottom-left';  
        } else if ( row === 1 ) {  
            return 'bottom';  
        } else {  
            return 'left';  
        }  
    } else {  
        if ( row === 0 ) {  
            return 'center';  
        } else if ( row === 1 ) {  
            return 'bottom-center';  
        } else {  
            return 'center';  
        }  
    }  
}  
  
function <function>( ...args ) {  
    const [ row, col ] = args;  
    if ( col === 0 ) {  
        if ( row === 0 ) {  
            return 'top-left';  
        } else if ( row === 1 ) {  
            return 'top';  
        } else {  
            return 'left';  
        }  
    } else if ( col === 1 ) {  
        if ( row === 0 ) {  
            return 'top-right';  
        } else if ( row === 1 ) {  
            return 'right';  
        } else {  
            return 'bottom-right';  
        }  
    } else if ( col === 2 ) {  
        if ( row === 0 ) {  
            return 'bottom-left';  
        } else if ( row === 1 ) {  
            return 'bottom';  
        } else {  
            return 'left';  
        }  
    } else {  
        if ( row === 0 ) {  
            return 'center';  
        } else if ( row === 1 ) {  
            return 'bottom-center';  
        } else {  
            return 'center';  
        }  
    }  
}  
  
function <function>( ...args ) {  
    const [ row, col ] = args;  
    if ( col === 0 ) {  
        if ( row === 0 ) {  
            return 'top-left';  
        } else if ( row === 1 ) {  
            return 'top';  
        } else {  
            return 'left';  
        }  
    } else if ( col === 1 ) {  
        if ( row === 0 ) {  
            return 'top-right';  
        } else if ( row === 1 ) {  
            return 'right';  
        } else {  
            return 'bottom-right';  
        }  
    } else if ( col === 2 ) {  
        if ( row === 0 ) {  
            return 'bottom-left';  
        } else if ( row === 1 ) {  
            return 'bottom';  
        } else {  
            return 'left';  
        }  
    } else {  
        if ( row === 0 ) {  
            return 'center';  
        } else if ( row === 1 ) {  
            return 'bottom-center';  
        } else {  
            return 'center';  
        }  
    }  
}
```

## THE COMPLETE JAVASCRIPT COURSE

SECTION

JAVASCRIPT IN THE BROWSER:  
DOM MANIPULATION AND  
EVENTS

LECTURE

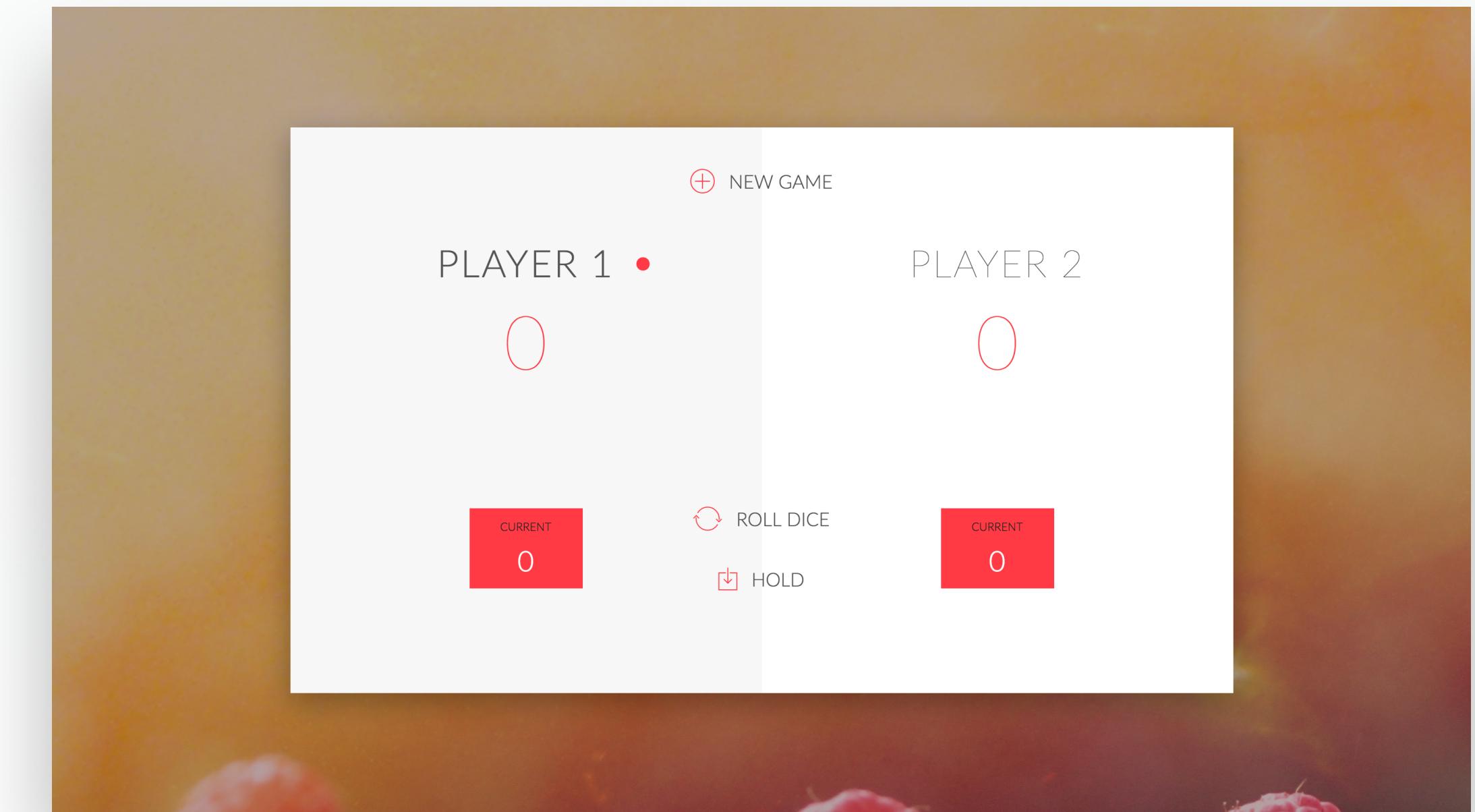
CREATING A GAME INITIALIZATION  
FUNCTION



**Jonas Schmedtmann**

# WHAT YOU WILL LEARN IN THIS LECTURE

- Practice, practice, practice...





**coding**heroes.io

## THE COMPLETE JAVASCRIPT COURSE

SECTION

JAVASCRIPT IN THE BROWSER:  
DOM MANIPULATION AND  
EVENTS

LECTURE

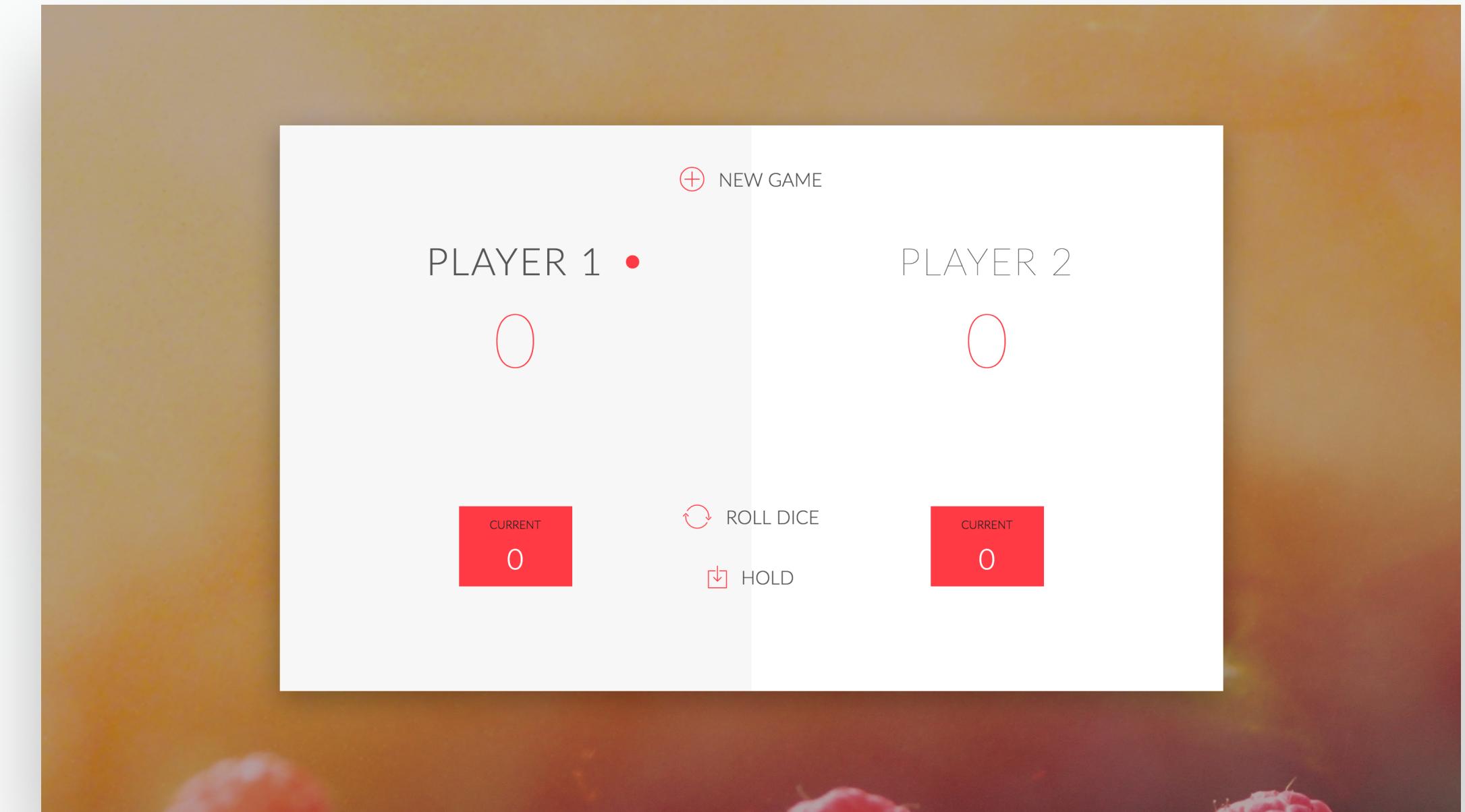
FINISHING TOUCHES



**Jonas Schmedtmann**

# WHAT YOU WILL LEARN IN THIS LECTURE

- What a state variable is, how to use it and why



# SECTION 5 – ADVANCED JAVASCRIPT: OBJECTS AND FUNCTIONS



## THE COMPLETE JAVASCRIPT COURSE

SECTION  
ADVANCED JAVASCRIPT: OBJECTS  
AND FUNCTIONS

LECTURE  
EVERYTHING IS AN OBJECT:  
INHERITANCE AND THE  
PROTOTYPE CHAIN



**Jonas Schmedtmann**

# OBJECTS IN JAVASCRIPT

## PRIMITIVES

- Numbers
- Strings
- Booleans
- Undefined
- Null

*Everything is an object.*

(Well, almost everything)

## EVERYTHING ELSE ...

- Arrays
- Functions
- Objects
- Dates
- Wrappers for Numbers, Strings, Booleans

... IS AN OBJECT

# THE OBJECT ORIENTED PARADIGM

## OBJECT-ORIENTED PROGRAMMING

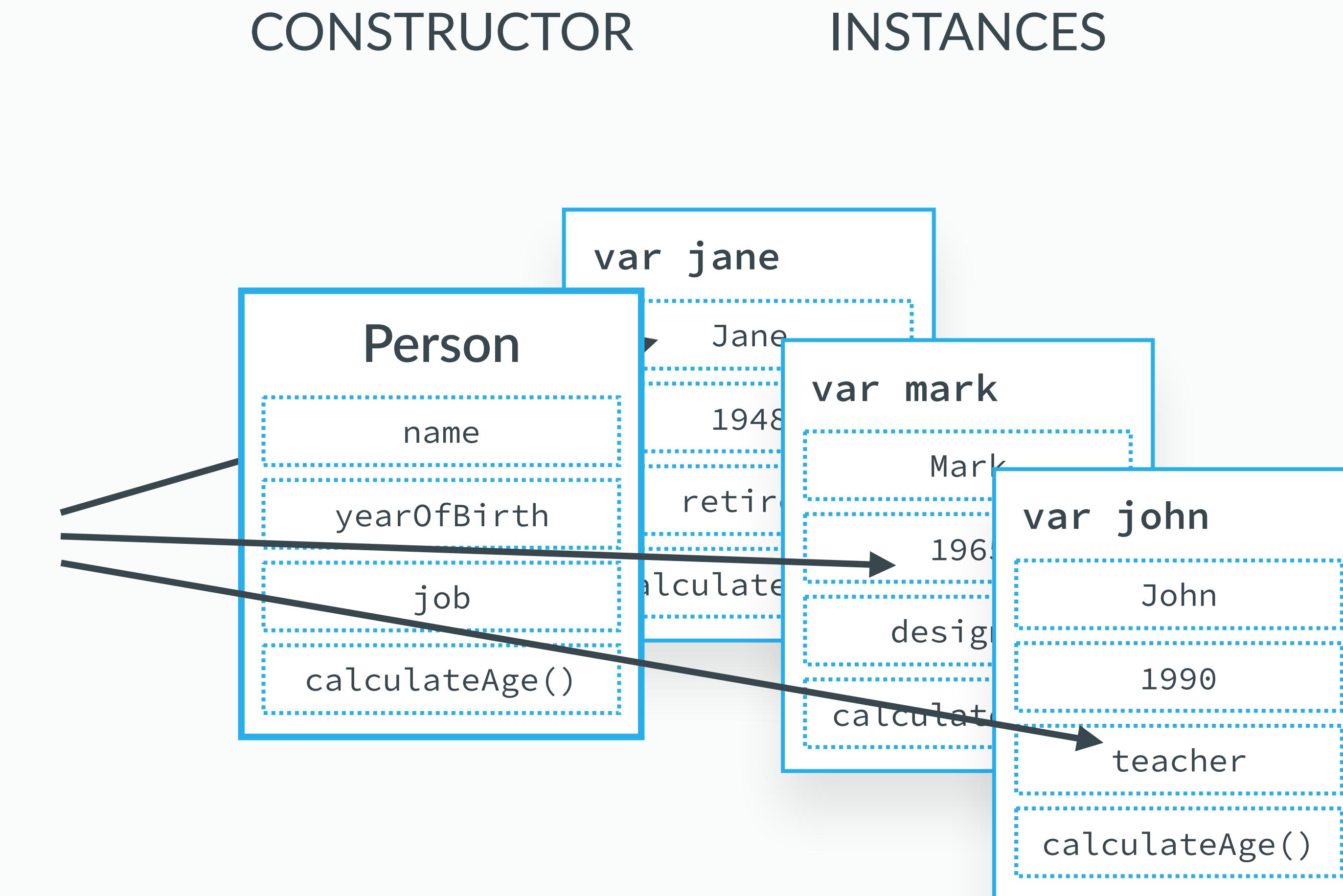
- Objects interacting with one another through methods and properties;
- Used to store data, structure applications into modules and keeping code clean.

```
var john = {  
    name: 'John',  
    yearOfBirth: 1990,  
    isMarried: false  
};
```

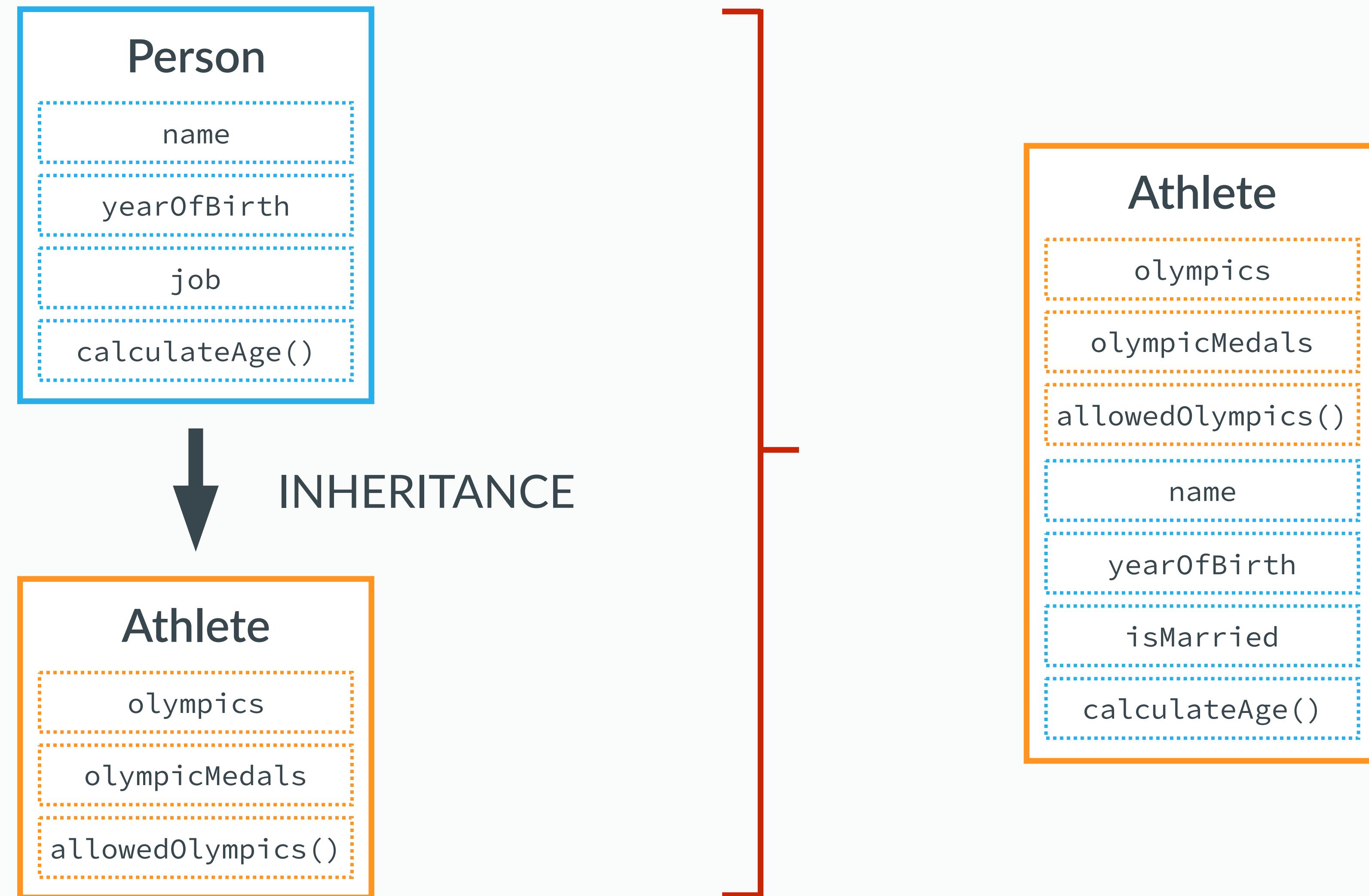
```
var jane = {  
    name: 'Jane',  
    yearOfBirth: 1969,  
    isMarried: true  
};
```

```
var mark = {  
    name: 'Mark',  
    yearOfBirth: 1948,  
    isMarried: true  
};
```

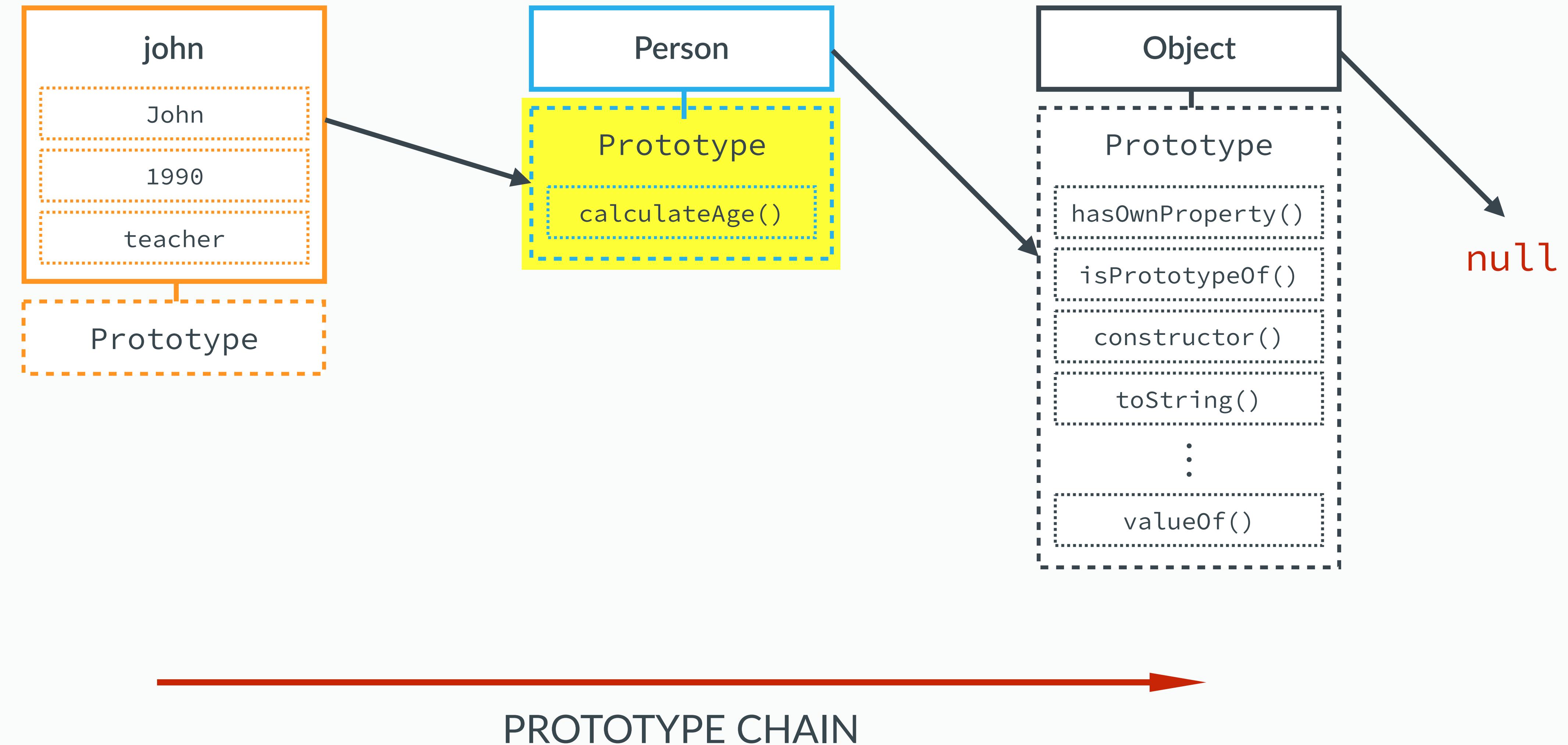
# CONSTRUCTORS AND INSTANCES IN JAVASCRIPT



# INHERITANCE IN GENERAL



# INHERITANCE IN JAVASCRIPT: PROTOTYPES AND PROTOTYPE CHAINS



# SUMMARY

- Every JavaScript object has a **prototype property**, which makes inheritance possible in JavaScript;
- The prototype property of an object is where we put methods and properties that we want **other objects to inherit**;
- The Constructor's prototype property is **NOT** the prototype of the Constructor itself, it's the prototype of **ALL** instances that are created through it;
- When a certain method (or property) is called, the search starts in the object itself, and if it cannot be found, the search moves on to the object's prototype. This continues until the method is found: **prototype chain**.



 codingheroes.io

# THE COMPLETE JAVASCRIPT AND Course

# SECTION

# ADVANCED JAVASCRIPT: OBJECTS AND FUNCTIONS

# LECTURE

## FIRST CLASS FUNCTIONS: PASSING FUNCTIONS AS ARGUMENTS



# Jonas Schmedtmann

# FUNCTIONS ARE ALSO OBJECTS IN JAVASCRIPT

- A function is an instance of the Object type;
- A function behaves like any other object;
- We can store functions in a variable;
- We can pass a function as an argument to another function;
- We can return a function from a function.



FIRST-CLASS FUNCTIONS



codingheroes.io

# THE COMPLETE JAVASCRIPT COURSE

## SECTION ADVANCED JAVASCRIPT: OBJECTS AND FUNCTIONS

### LECTURE CLOSURES



**Jonas Schmedtmann**

# CLOSURES

## CLOSURES SUMMARY

*An inner function has always access to the variables and parameters of its outer function, even after the outer function has returned.*

# HOW CLOSURES WORK



# CLOSURES

## CLOSURES SUMMARY (AGAIN :)

*An inner function has always access to the variables and parameters of its outer function, even after the outer function has returned.*

# SECTION 6 – PUTTING IT ALL TOGETHER: THE BUDGET APP PROJECT



**coding**heroes.io

```
function( 'post' );  
function( $ ) {  
    var $table = $( '#table' );  
  
    if( $table.length > 0 ) {  
        $table[0].tBodies[0].rows[0].cells[0].innerHTML = 'Last';  
  
        var $tbody = $table[0].tBodies[0];  
        var $tr = $tbody[0].rows[0];  
        var $td = $tr[0].cells[0];  
  
        if( $td.innerHTML === 'Last' ) {  
            $td.innerHTML = wp_get_attachment_link( $tr[0].cells[0].innerHTML );  
            $td.title = esc_attr( implode( ' ', $tr[0].cells[0].title ));  
            $td.title = esc_attr( get_the_title() );  
        }  
    }  
}  
wp_get_attachment_link = wp_get_attachment_link;  
wp_get_attachment_title = esc_attr( implode( ' ', wp_get_attachment_title() ));  
wp_get_attachment_title = esc_attr( get_the_title() );  
wp_get_attachment_size = wp_get_attachment_size;  
wp_get_attachment_size = wp_get_attachment_size;  
wp_get_attachment_size = wp_get_attachment_size;
```

## THE COMPLETE JAVASCRIPT COURSE

SECTION

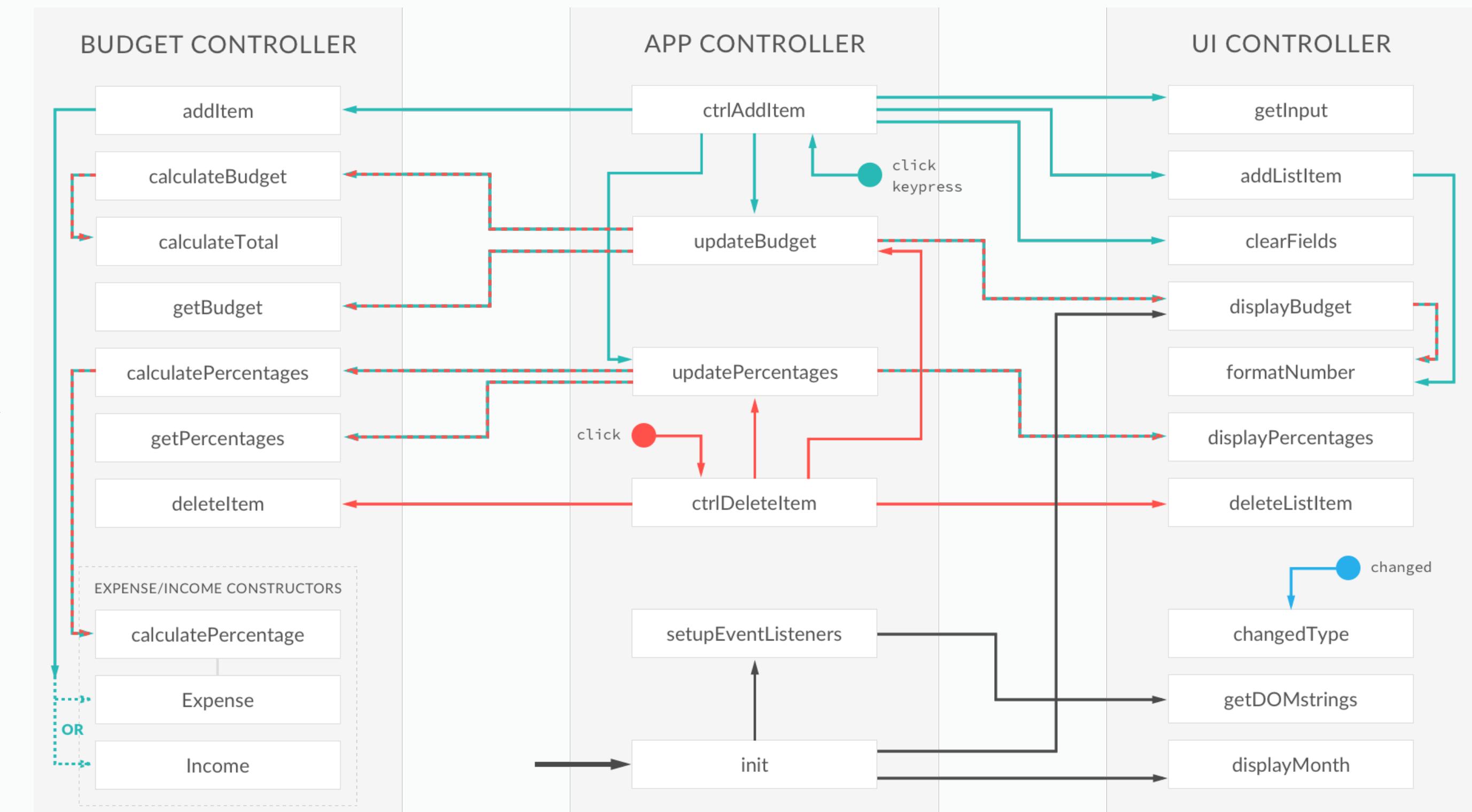
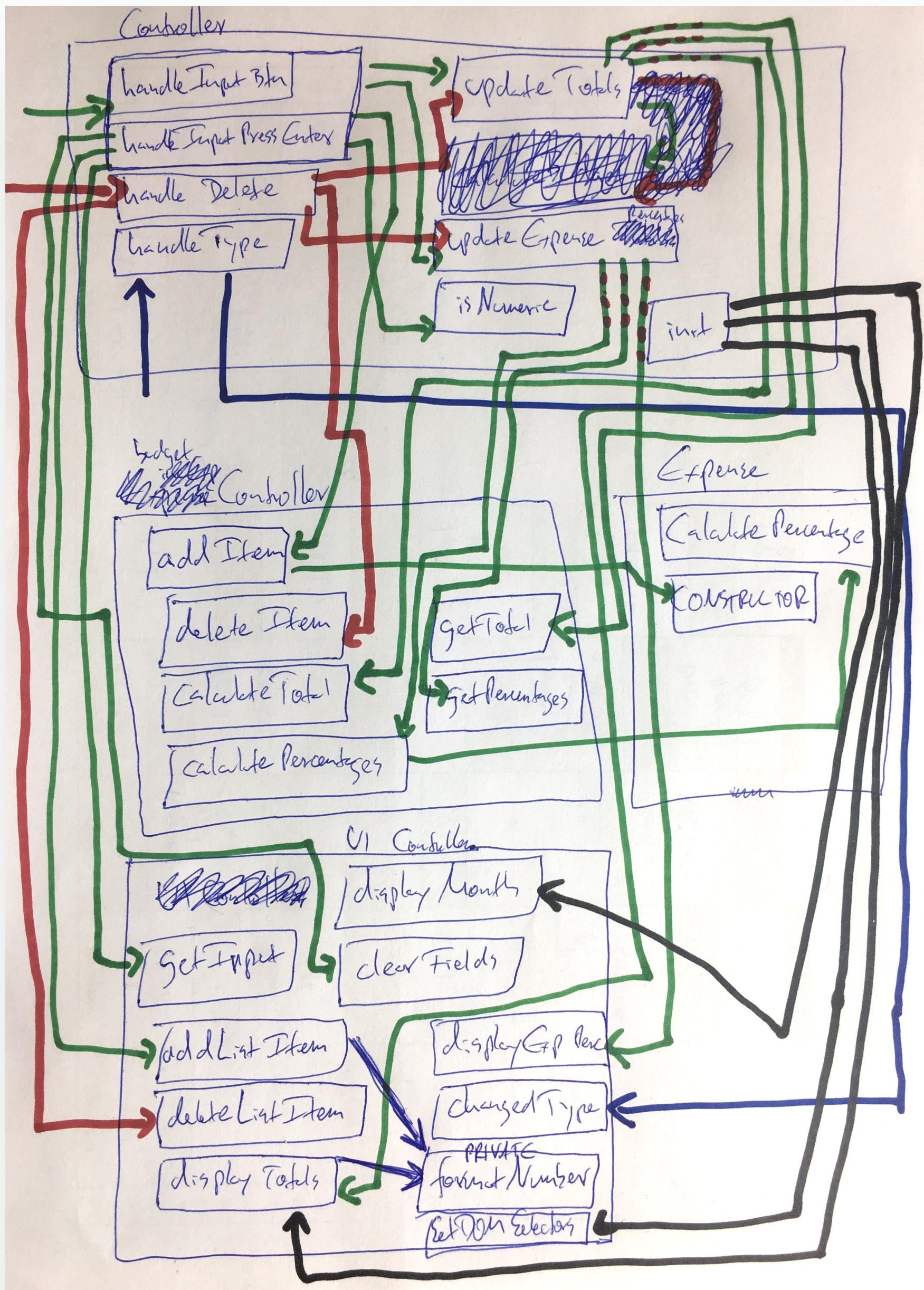
PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

LECTURE

PROJECT PLANNING AND  
ARCHITECTURE: STEP 1



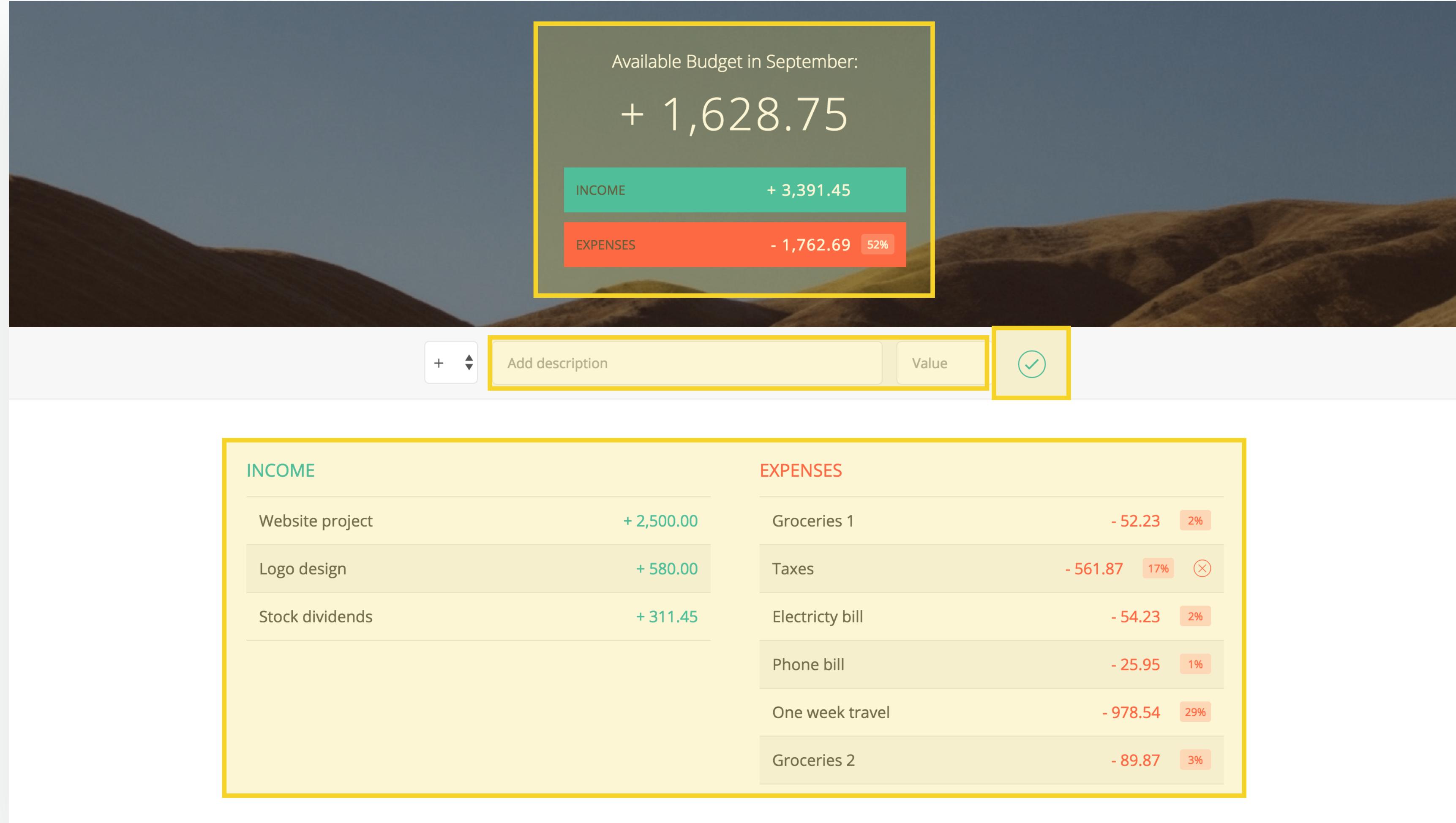
**Jonas Schmedtmann**



NOT SO FAST...

(LET'S THINK TOGETHER)

# PLANNING: STEP 1



## TO-DO LIST

Add event handler

Get input values

Add the new item to our data structure

Add the new item to the UI

Calculate budget

Update the UI

# STRUCTURING OUR CODE WITH MODULES

## MODULES

- Important aspect of any robust application's architecture;
- Keep the units of code for a project both cleanly separated and organized;
- Encapsulate some data into privacy and expose other data publicly.

# STRUCTURING OUR CODE WITH MODULES

UI MODULE

DATA MODULE

CONTROLLER MODULE

TO-DO LIST

Add event handler

Get input values

Add the new item to  
our data structure

Add the new item to  
the UI

Calculate budget

Update the UI



codingheroes.io

## THE COMPLETE JAVASCRIPT COURSE

SECTION

PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

LECTURE

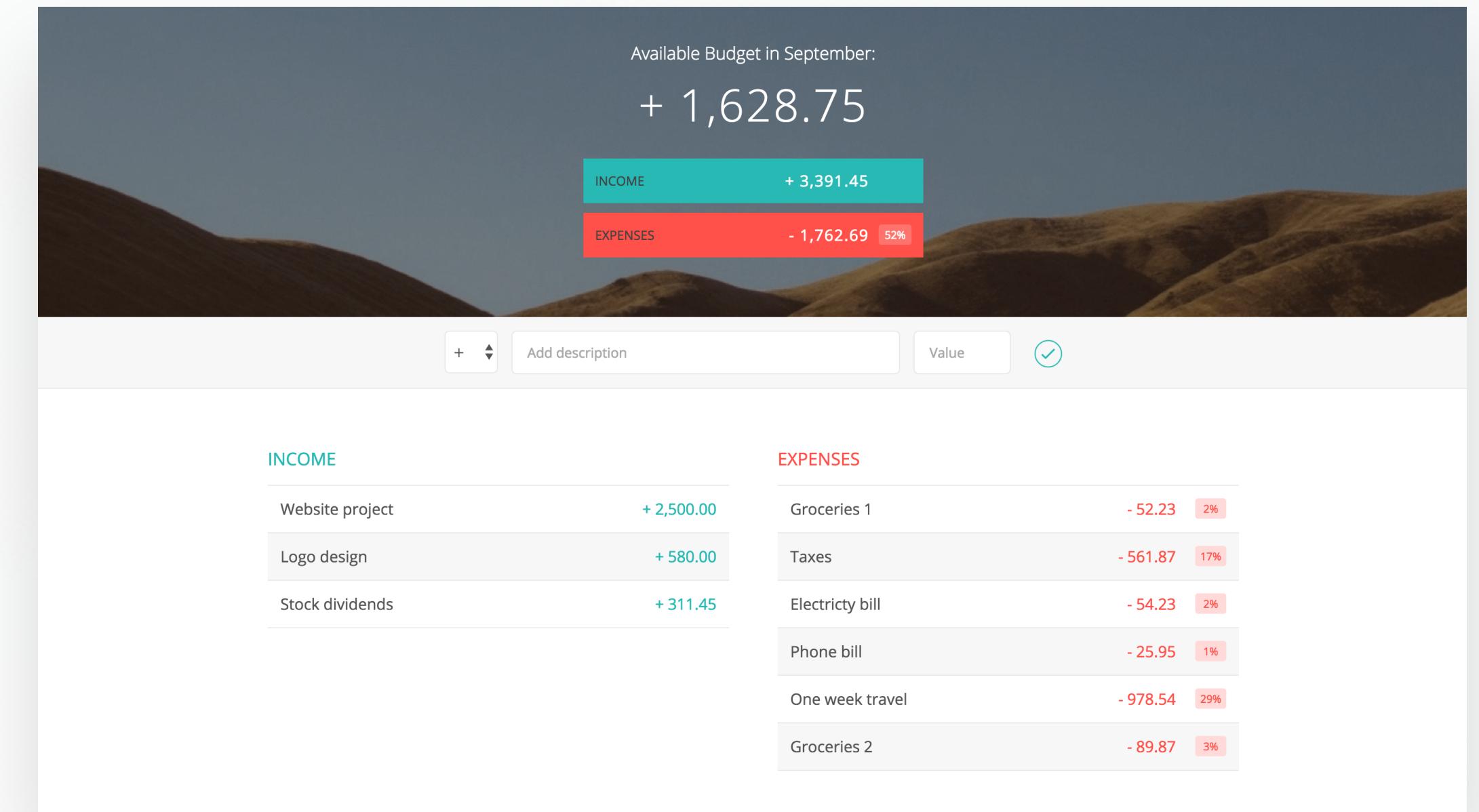
IMPLEMENTING THE MODULE  
PATTERN



**Jonas Schmedtmann**

# WHAT YOU WILL LEARN IN THIS LECTURE

- How to use the module pattern;
- More about private and public data, encapsulation and separation of concerns.





codingheroes.io

## THE COMPLETE JAVASCRIPT COURSE

SECTION

PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

LECTURE

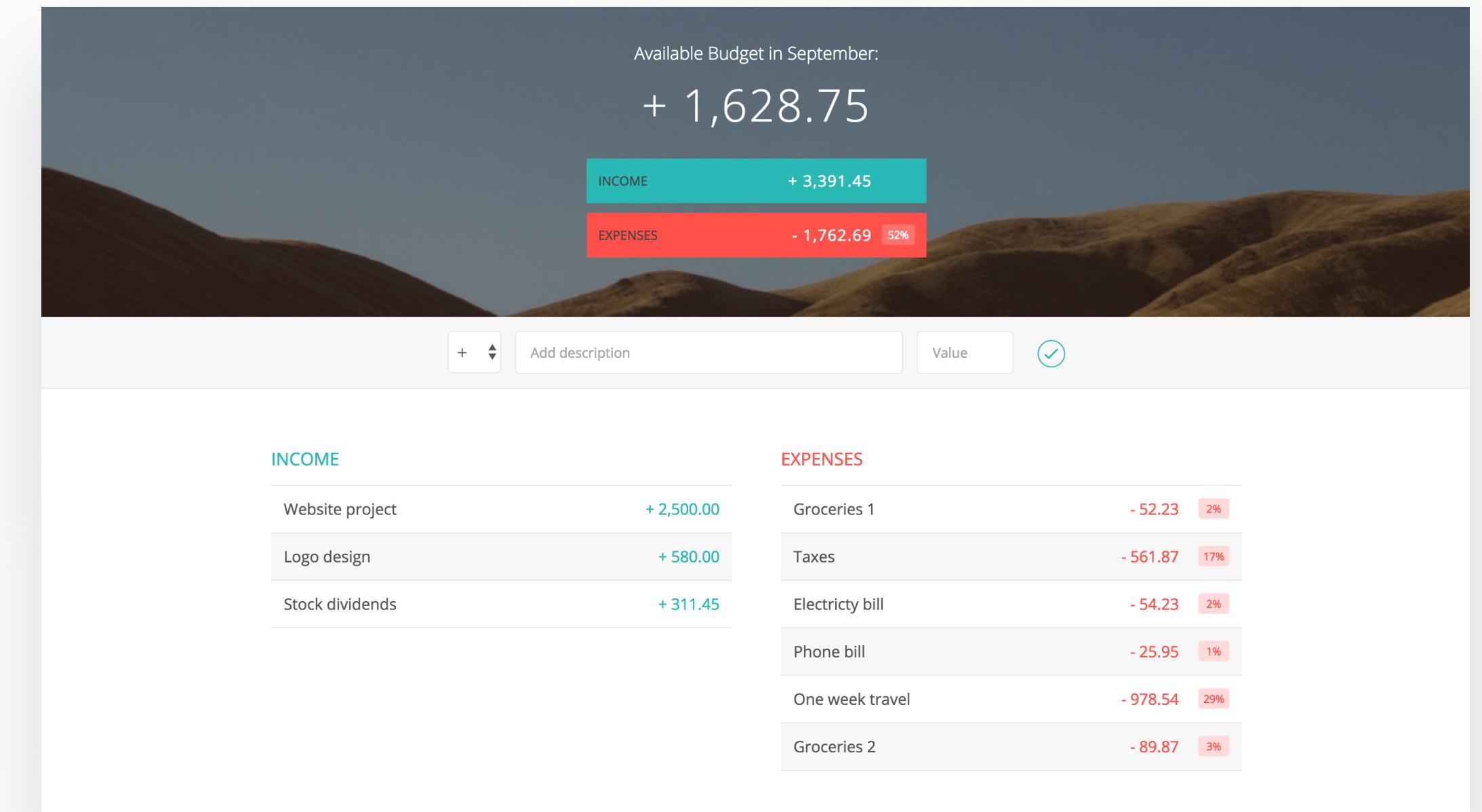
SETTING UP THE FIRST EVENT  
LISTENERS



Jonas Schmedtmann

# WHAT YOU WILL LEARN IN THIS LECTURE

- How to set up event listeners for keypress events;
- How to use event object.





**coding**heroes.io

```
        <td></td>
    <tr>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
    </tr>
</table>
<div>
    
    <span>Read more...</span>
</div>
```

## THE COMPLETE JAVASCRIPT COURSE

SECTION

PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

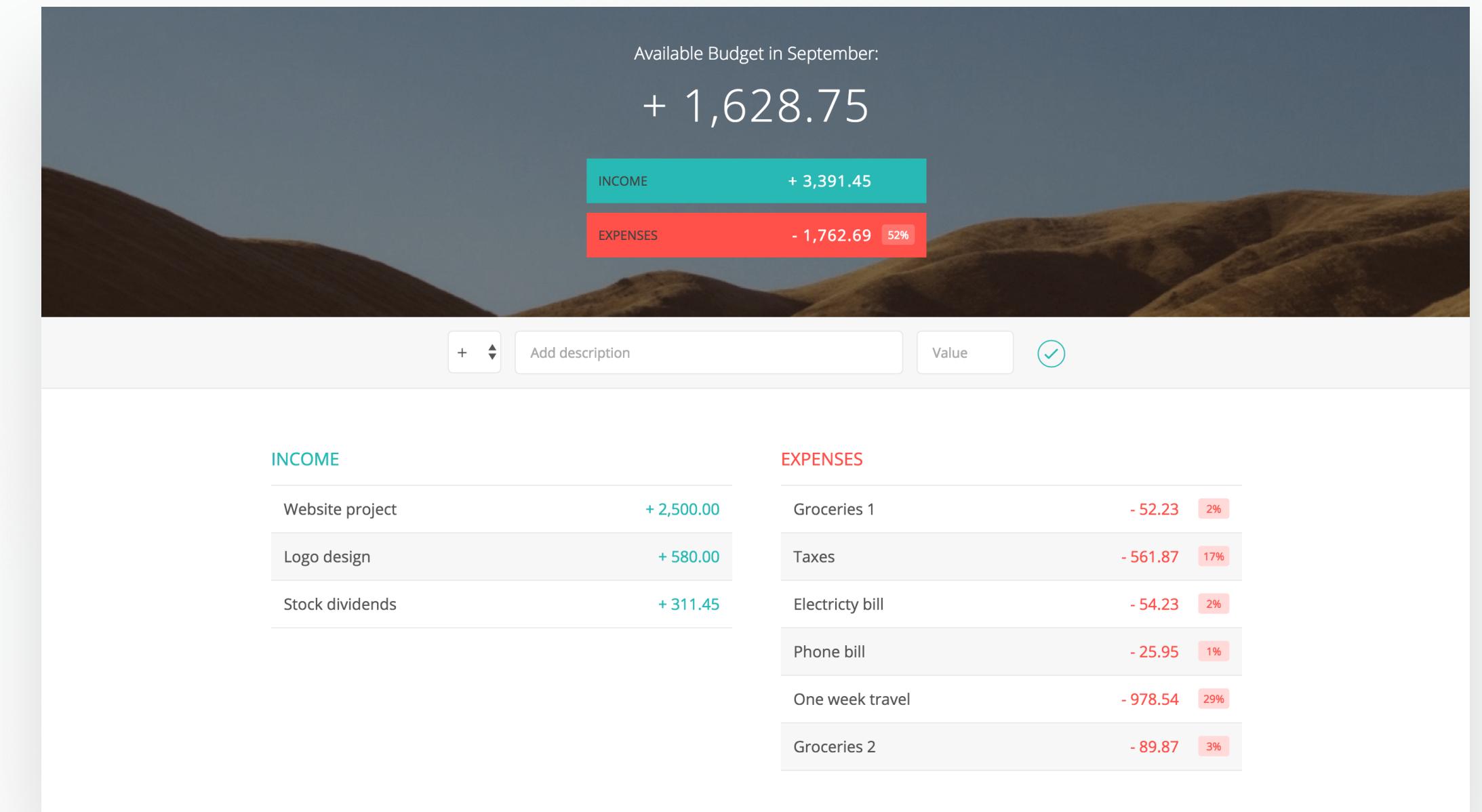
LECTURE  
READING INPUT DATA



**Jonas Schmedtmann**

# WHAT YOU WILL LEARN IN THIS LECTURE

- How to read data from different HTML input types.





**coding**heroes.io

```
function( 'post' );  
function( 'comment' );  
function( 'category' );  
function( 'tag' );  
function( 'attachment' );  
  
if ( $columns == 0 )  
    $columns = 1;  
else if ( $columns == 1 )  
    $columns = 2;  
  
$content = wp_get_attachment_image( $attachment_id, $size );  
$content .= wp_get_attachment_image_src( $attachment_id, $size );  
$content = esc_attr( implode( ' ', $content ) );  
$content = esc_attr( get_the_title() );  
  
return $content;
```

## THE COMPLETE JAVASCRIPT COURSE

SECTION

PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

LECTURE

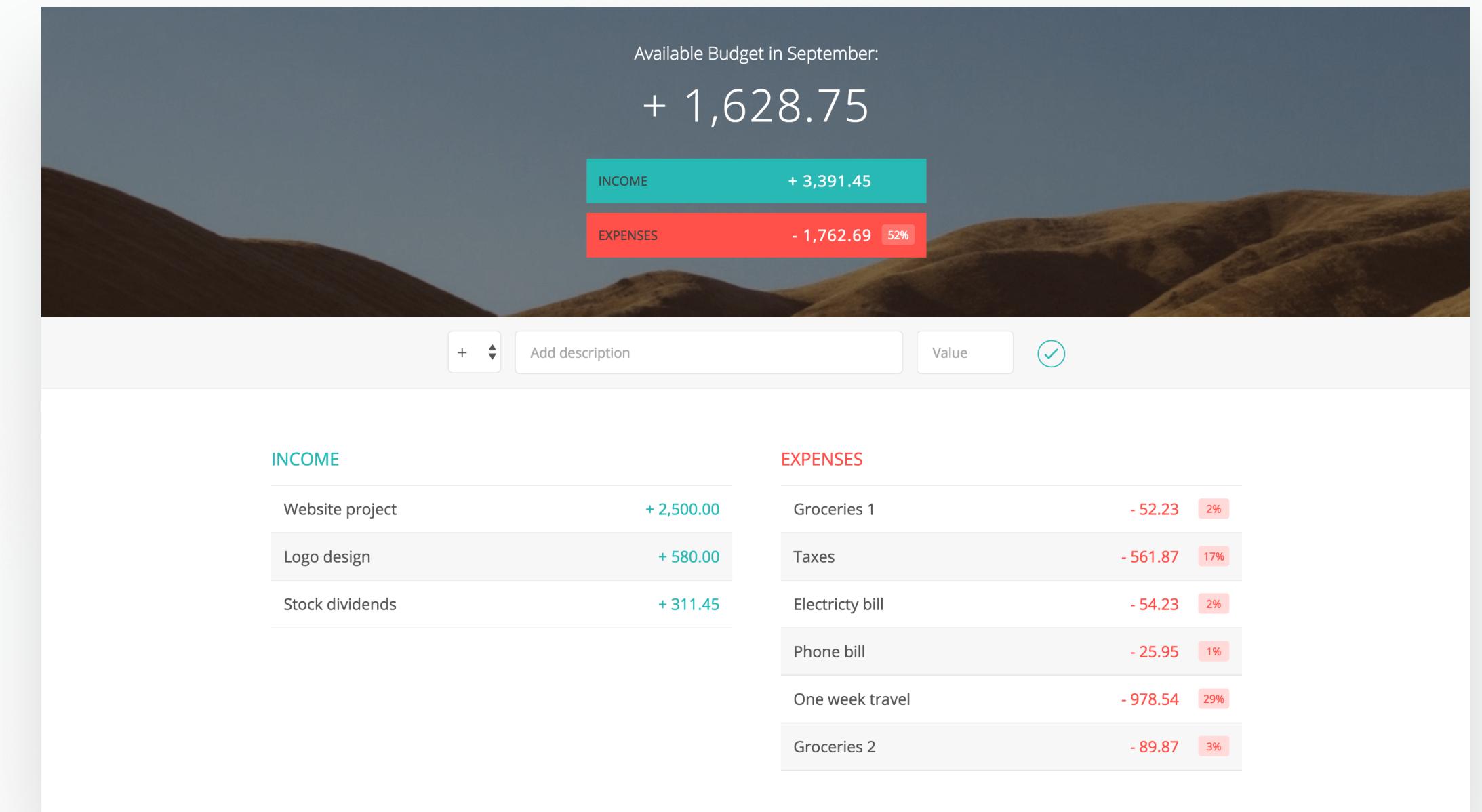
CREATING AN INITIALIZATION  
FUNCTION



**Jonas Schmedtmann**

# WHAT YOU WILL LEARN IN THIS LECTURE

- How and why to create an initialization function.





codingheroes.io

```
function <...> ( ...args ) {  
    const [ row, columns ] = args;  
    const [ first, ...rest ] = columns;  
    const last = rest[ rest.length - 1 ];  
  
    const image = first.attachment_url;  
    const link = first.image_link;  
  
    const attachment = wp_get_attachment;  
    const attrs = esc_attr( implode( ' ', [  
        'title' + esc_attr( get_the_title() ),  
        'id' + esc_attr( get_id() ),  
        'alt' + esc_attr( get_post_meta( first.ID, '_wp_attachment_image_alt', true ) ),  
        'src' + esc_attr( wp_get_attachment_image_src( first.ID, 'shop_thumbnail_size' )[ 0 ] ),  
        'size' + esc_attr( wp_get_attachment_image_srcset( first.ID, 'shop_thumbnail_size' )[ 0 ] ),  
        'width' + esc_attr( wp_get_attachment_image_srcset( first.ID, 'shop_thumbnail_size' )[ 1 ] ),  
        'height' + esc_attr( wp_get_attachment_image_srcset( first.ID, 'shop_thumbnail_size' )[ 2 ] )  
    ] ) );  
    const title = esc_attr( get_the_title() );  
    const id = esc_attr( get_id() );  
    const alt = esc_attr( get_post_meta( first.ID, '_wp_attachment_image_alt', true ) );  
    const src = esc_attr( wp_get_attachment_image_src( first.ID, 'shop_thumbnail_size' )[ 0 ] );  
    const size = esc_attr( wp_get_attachment_image_srcset( first.ID, 'shop_thumbnail_size' )[ 0 ] );  
    const width = esc_attr( wp_get_attachment_image_srcset( first.ID, 'shop_thumbnail_size' )[ 1 ] );  
    const height = esc_attr( wp_get_attachment_image_srcset( first.ID, 'shop_thumbnail_size' )[ 2 ] );  
}
```

## THE COMPLETE JAVASCRIPT COURSE

SECTION

PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

LECTURE

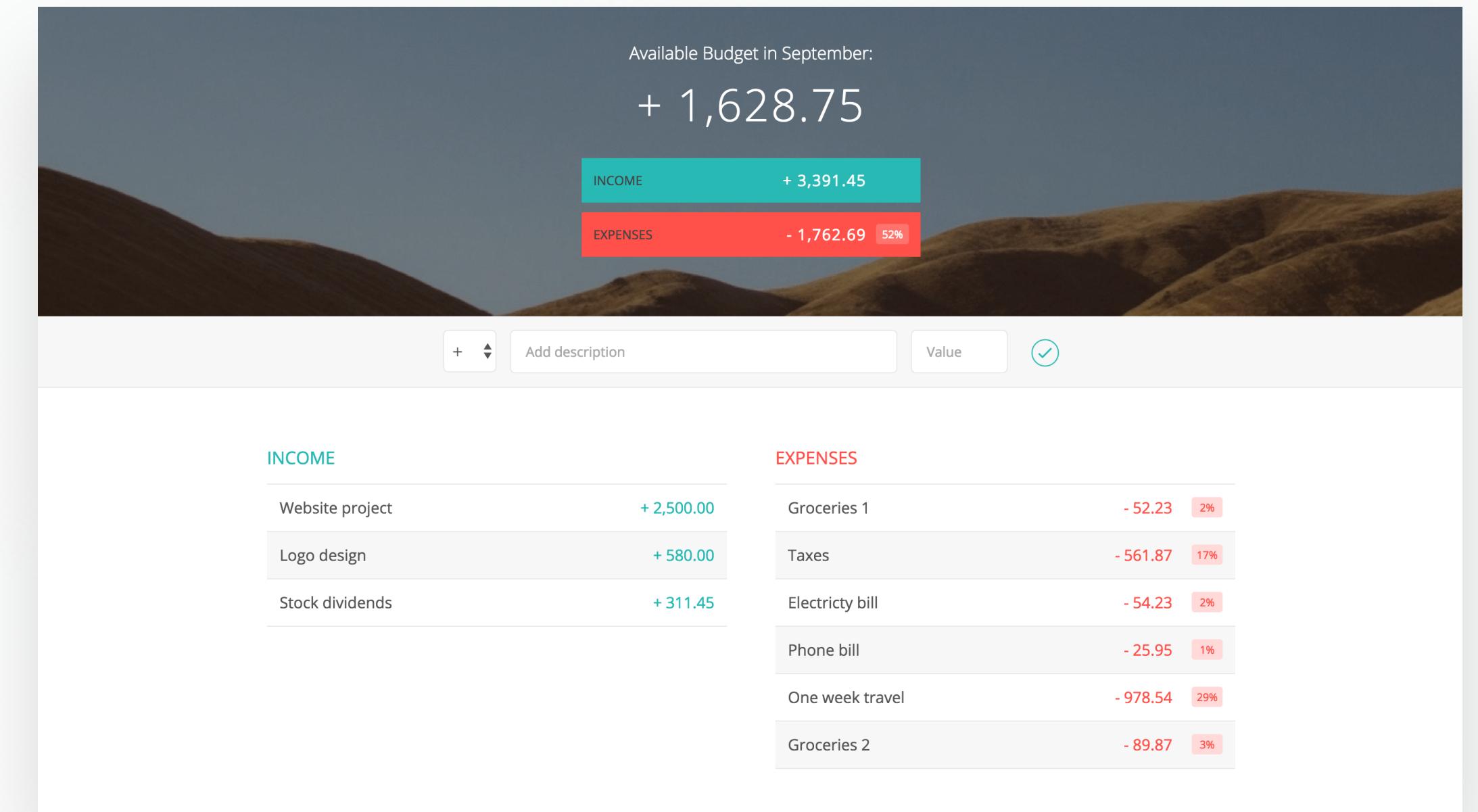
CREATING INCOME AND EXPENSE  
FUNCTION CONSTRUCTORS



Jonas Schmedtmann

# WHAT YOU WILL LEARN IN THIS LECTURE

- How to choose function constructors that meet our application's needs;
- How to set up a proper data structure for our budget controller.





codingheroes.io

## THE COMPLETE JAVASCRIPT COURSE

SECTION

PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

LECTURE

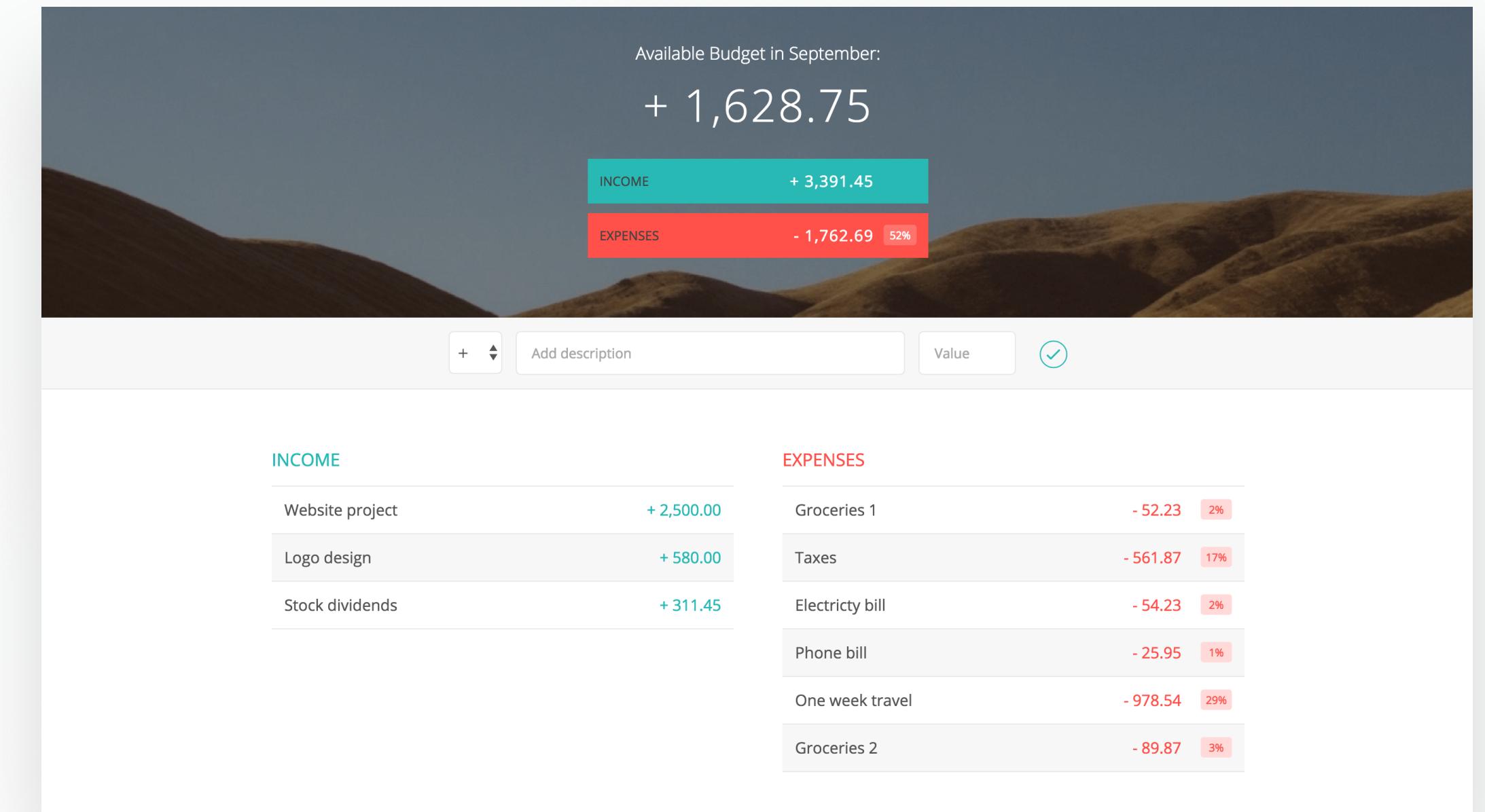
ADDING A NEW ITEM TO OUR  
BUDGET CONTROLLER



**Jonas Schmedtmann**

# WHAT YOU WILL LEARN IN THIS LECTURE

- How to avoid conflicts in our data structures;
- How and why to pass data from one module to another.





codingheroes.io

```
function( 'post' );  
function( 'comment' );  
function( 'category' );  
function( 'tag' );  
  
if( $row == 1 ) $columns = 0;  
else $columns = 1;  
$row++;  
  
if( $row == 1 ) $last = true;  
else $last = false;  
  
if( $row == 1 ) $image_link = true;  
else $image_link = false;  
  
if( $row == 1 ) $attachment = wp_get_attachment_image();  
else $attachment = esc_attr( implode( '  
', $attachment ) );  
$attachment = esc_attr( get_the_title() );  
  
if( $row == 1 ) $slide = true;  
else $slide = false;  
  
if( $row == 1 ) $easyzoom = true;  
else $easyzoom = false;  
$easyzoom = esc_attr( implode( '  
', $easyzoom ) );  
$easyzoom = esc_attr( get_the_title() );  
  
if( $row == 1 ) $wp_get_attachment = true;  
else $wp_get_attachment = false;  
$wp_get_attachment = esc_attr( implode( '  
', $wp_get_attachment ) );  
$wp_get_attachment = esc_attr( get_the_title() );  
  
if( $row == 1 ) $wp_get_attachment_size = true;  
else $wp_get_attachment_size = false;  
$wp_get_attachment_size = esc_attr( implode( '  
', $wp_get_attachment_size ) );  
$wp_get_attachment_size = esc_attr( get_the_title() );  
  
if( $row == 1 ) $shop_thumbnail_size = true;  
else $shop_thumbnail_size = false;  
$shop_thumbnail_size = esc_attr( implode( '  
', $shop_thumbnail_size ) );  
$shop_thumbnail_size = esc_attr( get_the_title() );  
  
if( $row == 1 ) $slide = true;  
else $slide = false;
```

## THE COMPLETE JAVASCRIPT COURSE

SECTION

PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

LECTURE

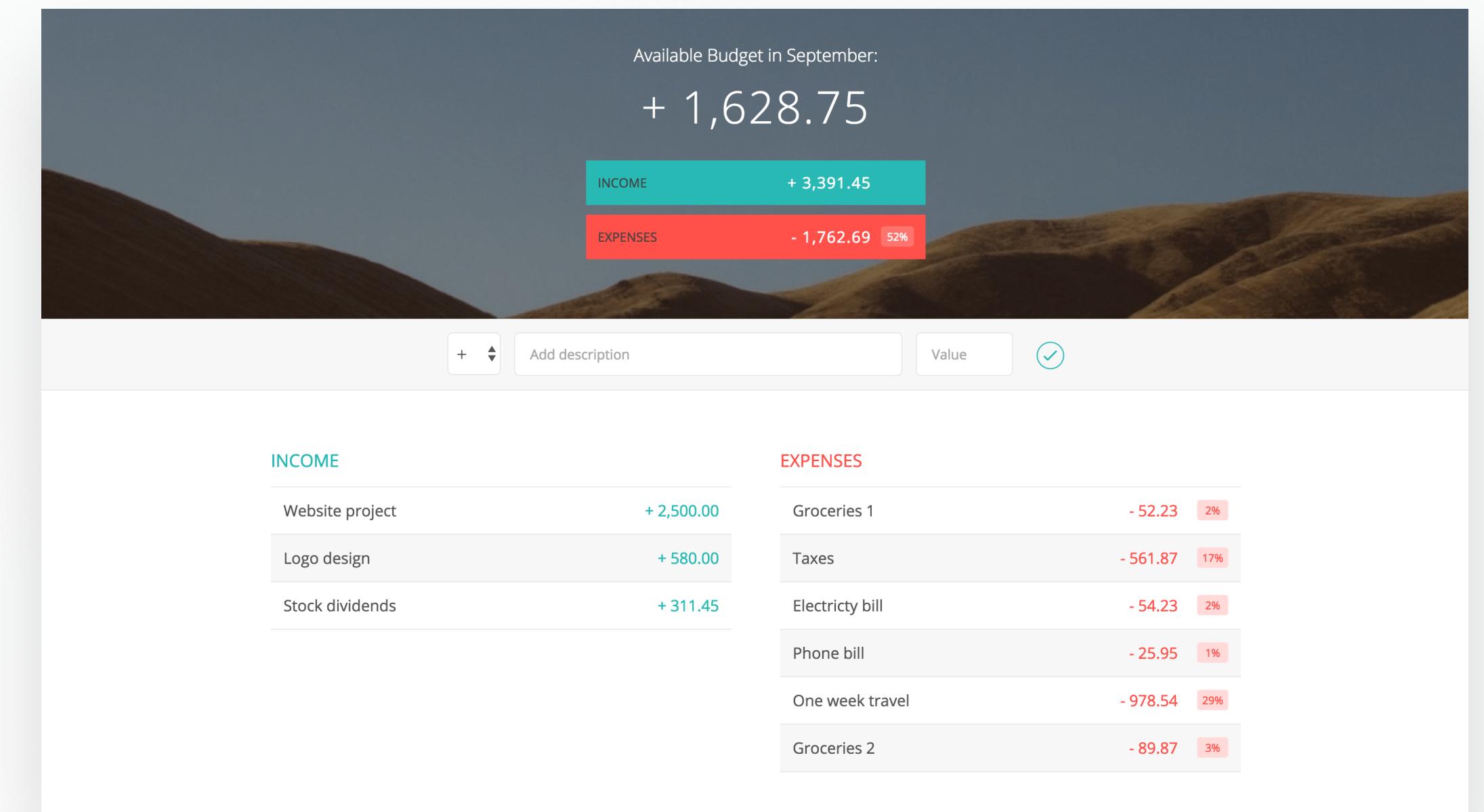
ADDING A NEW ITEM TO THE UI



Jonas Schmedtmann

# WHAT YOU WILL LEARN IN THIS LECTURE

- A technique for adding big chunks of HTML into the DOM;
- How to replace parts of strings;
- How to do DOM manipulation using the `insertAdjacentHTML` method.





codingheroes.io

## THE COMPLETE JAVASCRIPT COURSE

SECTION

PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

LECTURE

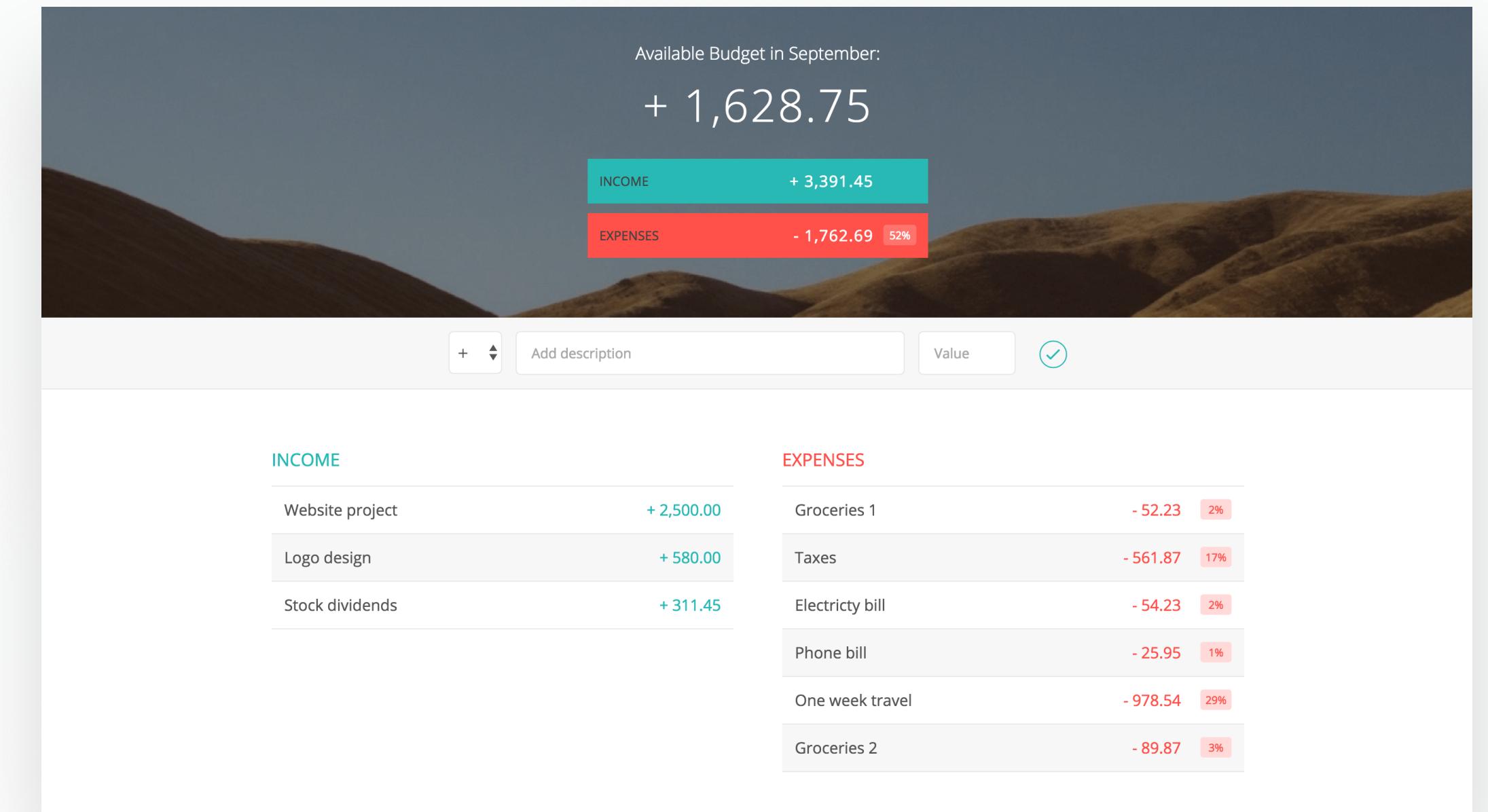
CLEARING OUR INPUT FIELDS



Jonas Schmedtmann

# WHAT YOU WILL LEARN IN THIS LECTURE

- How to clear HTML fields;
- How to use querySelectorAll,
- How to convert a list to an array;
- A better way to loop over an array  
then for loops: foreach.





codingheroes.io

## THE COMPLETE JAVASCRIPT COURSE

SECTION

PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

LECTURE

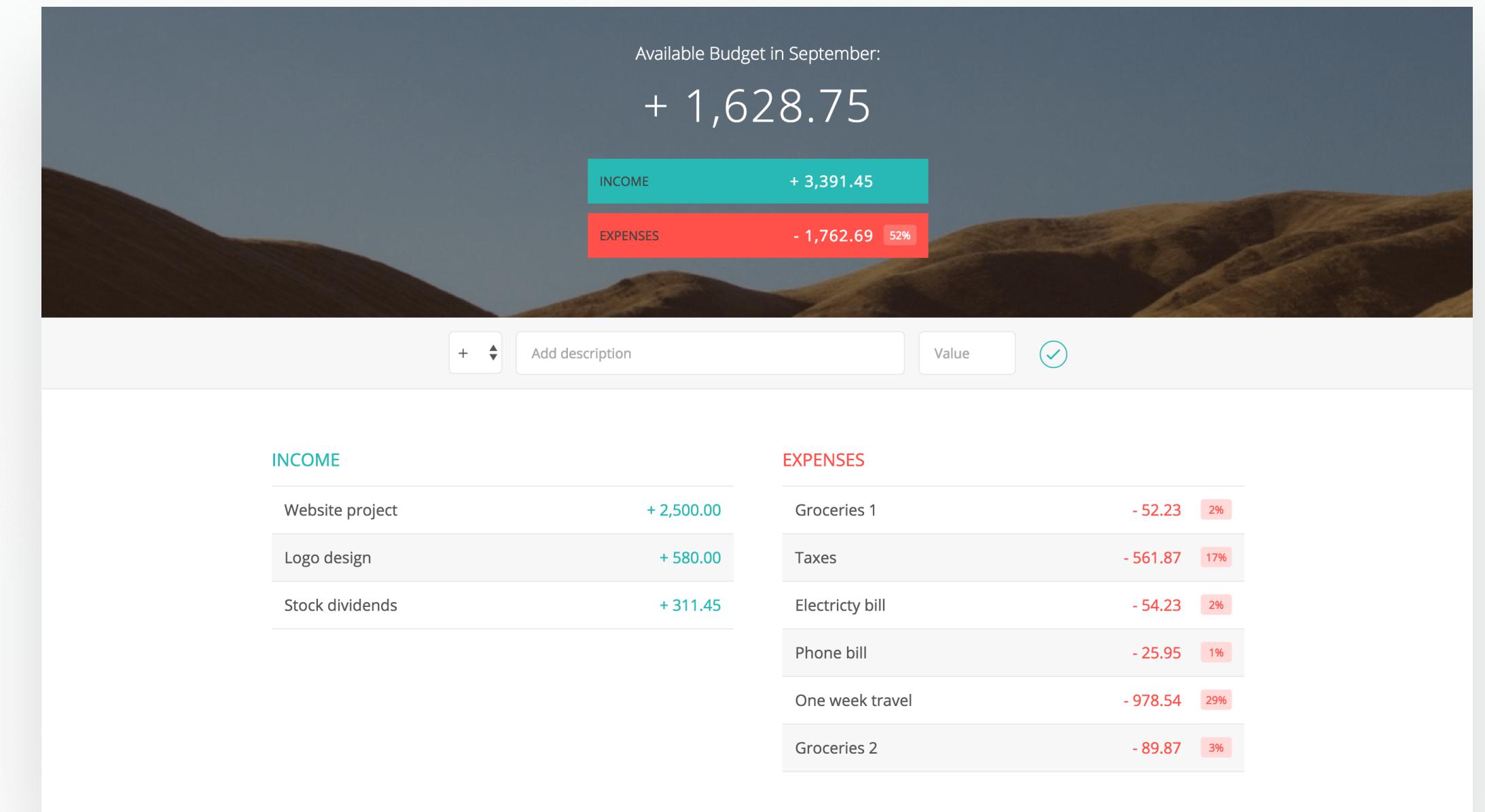
UPDATING THE BUDGET:  
CONTROLLER



Jonas Schmedtmann

# WHAT YOU WILL LEARN IN THIS LECTURE

- How to convert field inputs to numbers;
- How to prevent false inputs.





**coding**heroes.io

## THE COMPLETE JAVASCRIPT COURSE

SECTION

PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

LECTURE

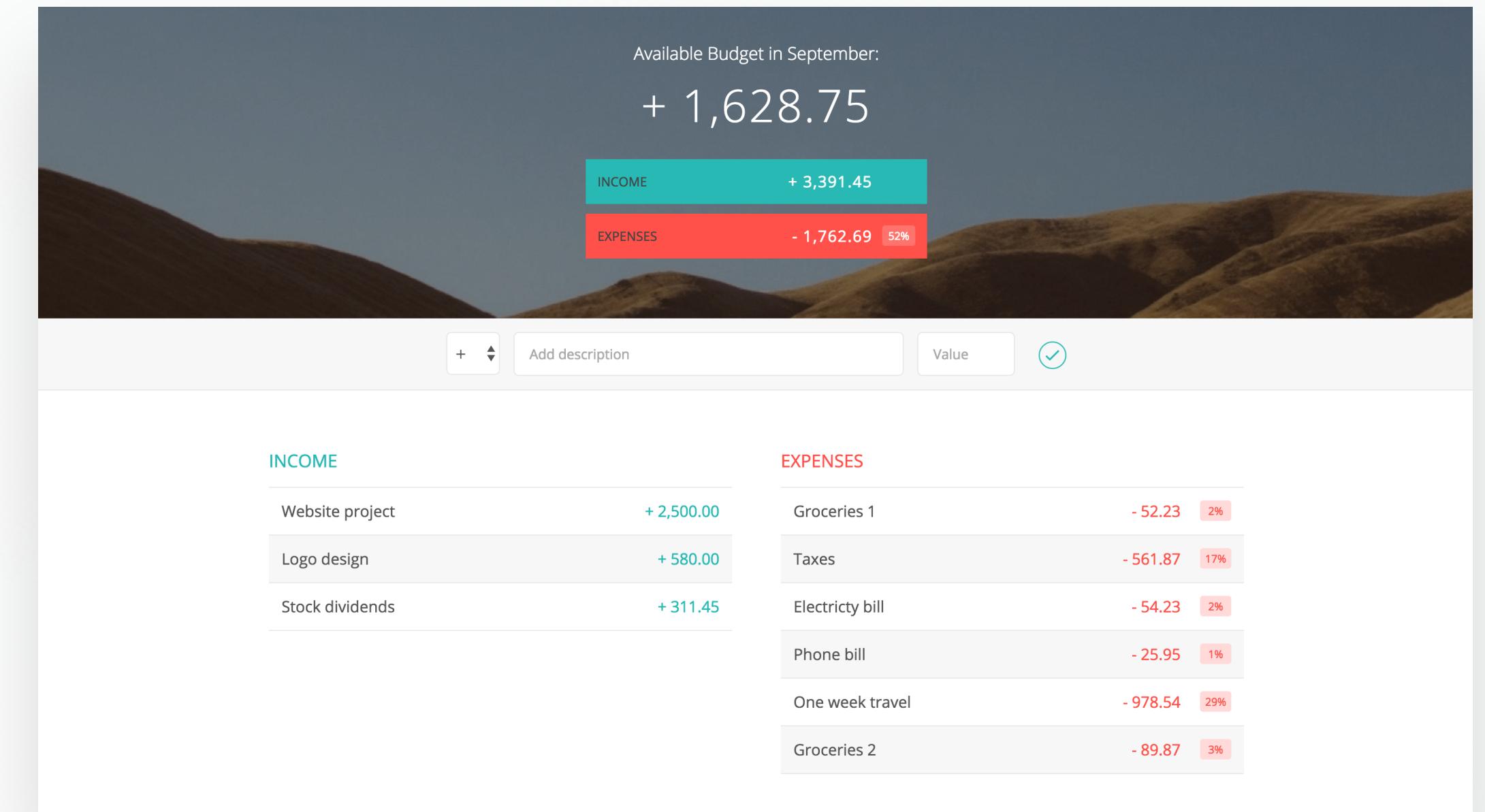
UPDATING THE BUDGET: BUDGET  
CONTROLLER



**Jonas Schmedtmann**

# WHAT YOU WILL LEARN IN THIS LECTURE

- How and why to create simple, reusable functions with only one purpose;
- How to sum all elements of an array using the `forEach` method.





codingheroes.io

```
function( 'post' );  
function( $ ) {  
    var $grid = $('#grid');  
    var $table = $('#table');  
    var $tbody = $table.find('tbody');  
    var $thead = $table.find('thead');  
    var $tfoot = $table.find('tfoot');  
    var $tbodyRows = $tbody.find('tr');  
    var $tbodyCells = $tbodyRows.find('td');  
    var $theadCells = $thead.find('th');  
    var $tfootCells = $tfoot.find('th');  
    var $gridRow = $grid.find('.row');  
    var $gridCell = $gridRow.find('.cell');  
    var $gridImage = $gridRow.find('.image');  
    var $gridCaption = $gridRow.find('.caption');  
    var $gridLink = $gridRow.find('.link');  
  
    // Set the grid columns  
    $gridRow.append($gridImage);  
    $gridRow.append($gridCaption);  
    $gridRow.append($gridLink);  
  
    if ($gridRow.length > 1) {  
        $gridRow.last().append($gridImage);  
        $gridRow.last().append($gridCaption);  
        $gridRow.last().append($gridLink);  
    }  
  
    // Set the table columns  
    $tbodyCells.append($theadCells);  
    $tbodyCells.append($tfootCells);  
  
    // Set the table rows  
    $tbodyCells.each(function() {  
        var $cell = $(this);  
        var $row = $cell.parent();  
        var $rowIndex = $tbodyRows.index($row);  
        var $rowCount = $tbodyRows.length;  
        var $rowLast = $rowIndex === $rowCount - 1;  
        var $rowFirst = $rowIndex === 0;  
        var $rowMiddle = !$rowFirst & !$rowLast;  
  
        if ($rowLast) {  
            $cell.append($tfootCells);  
        } else if ($rowFirst) {  
            $cell.append($theadCells);  
        } else {  
            $cell.append($tbodyCells);  
        }  
    });  
  
    // Set the table header  
    $theadCells.append($gridCaption);  
    $theadCells.append($gridLink);  
  
    // Set the table footer  
    $tfootCells.append($gridCaption);  
    $tfootCells.append($gridLink);  
  
    // Set the table body  
    $tbodyCells.append($gridImage);  
    $tbodyCells.append($gridCaption);  
    $tbodyCells.append($gridLink);  
});  
});
```

## THE COMPLETE JAVASCRIPT COURSE

SECTION

PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

LECTURE

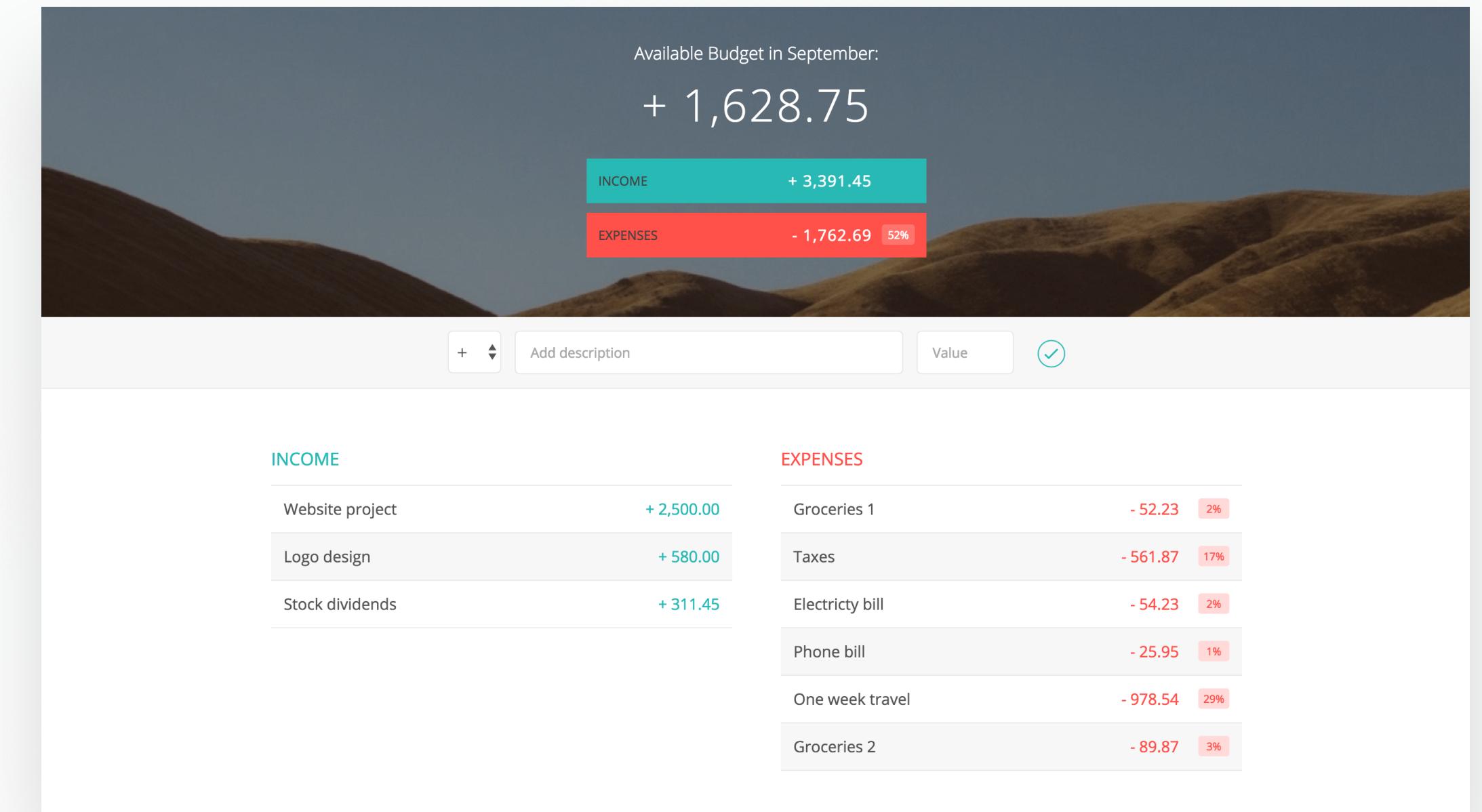
UPDATING THE BUDGET: UI  
CONTROLLER



Jonas Schmedtmann

# WHAT YOU WILL LEARN IN THIS LECTURE

- Practice DOM manipulation by updating the budget and total values.





**coding**heroes.io

```
function( 'post' );  
  
if( $post->post_type == 'attachment' ) {  
    if( $post->post_parent == 0 ) {  
        $post->post_parent = null;  
    }  
    $post->post_parent = wp_get_attachment_ur  
    $post->post_type = 'image';  
}  
  
$post->post_content = wp_get_attachment_content();  
$post->post_excerpt = esc_attr( implode( '  
    $post->post_title = esc_attr( get_the_title();  
  
    echo( '

## THE COMPLETE JAVASCRIPT COURSE



SECTION



PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT



LECTURE



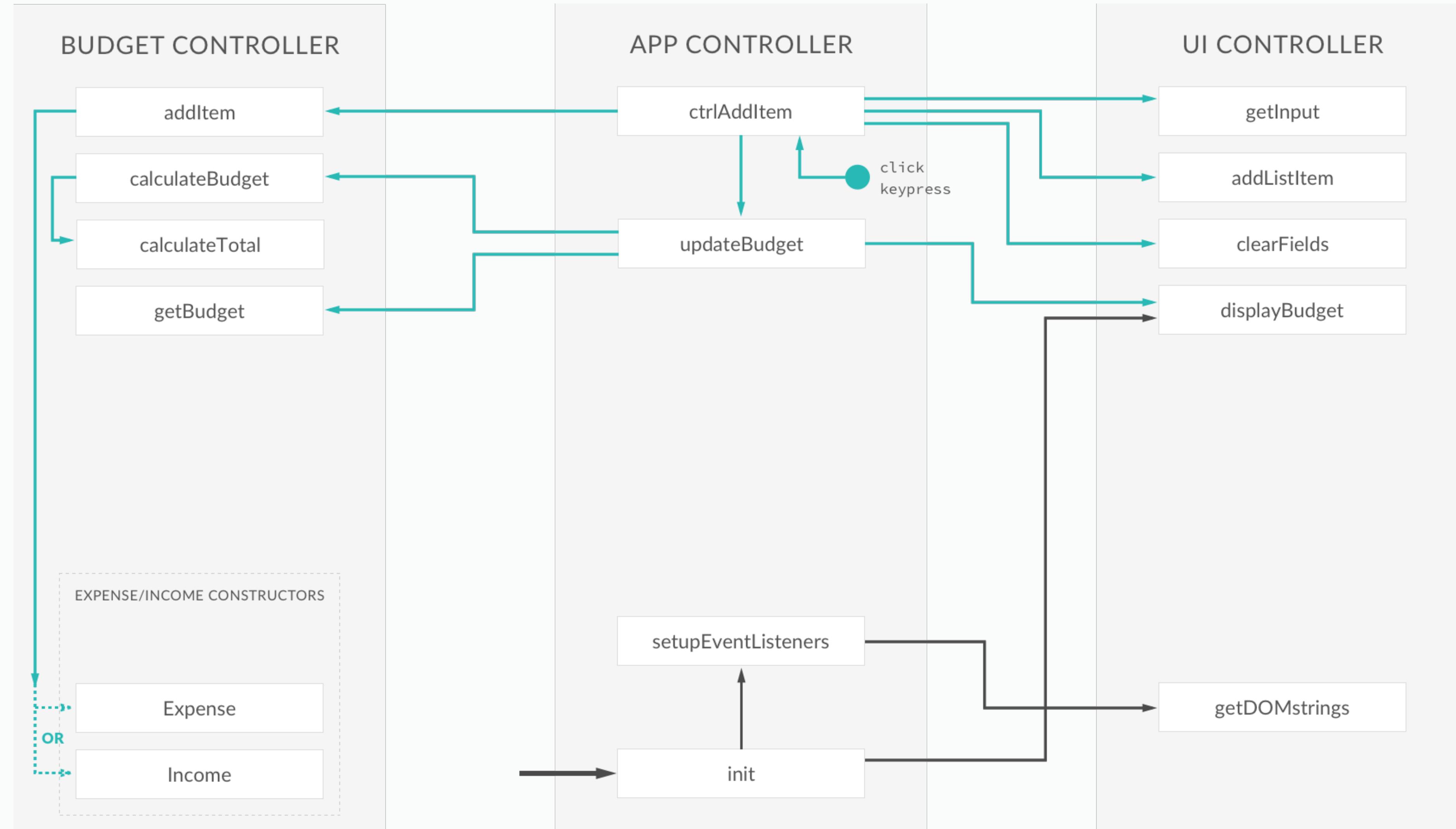
PROJECT PLANNING AND  
ARCHITECTURE: STEP 2



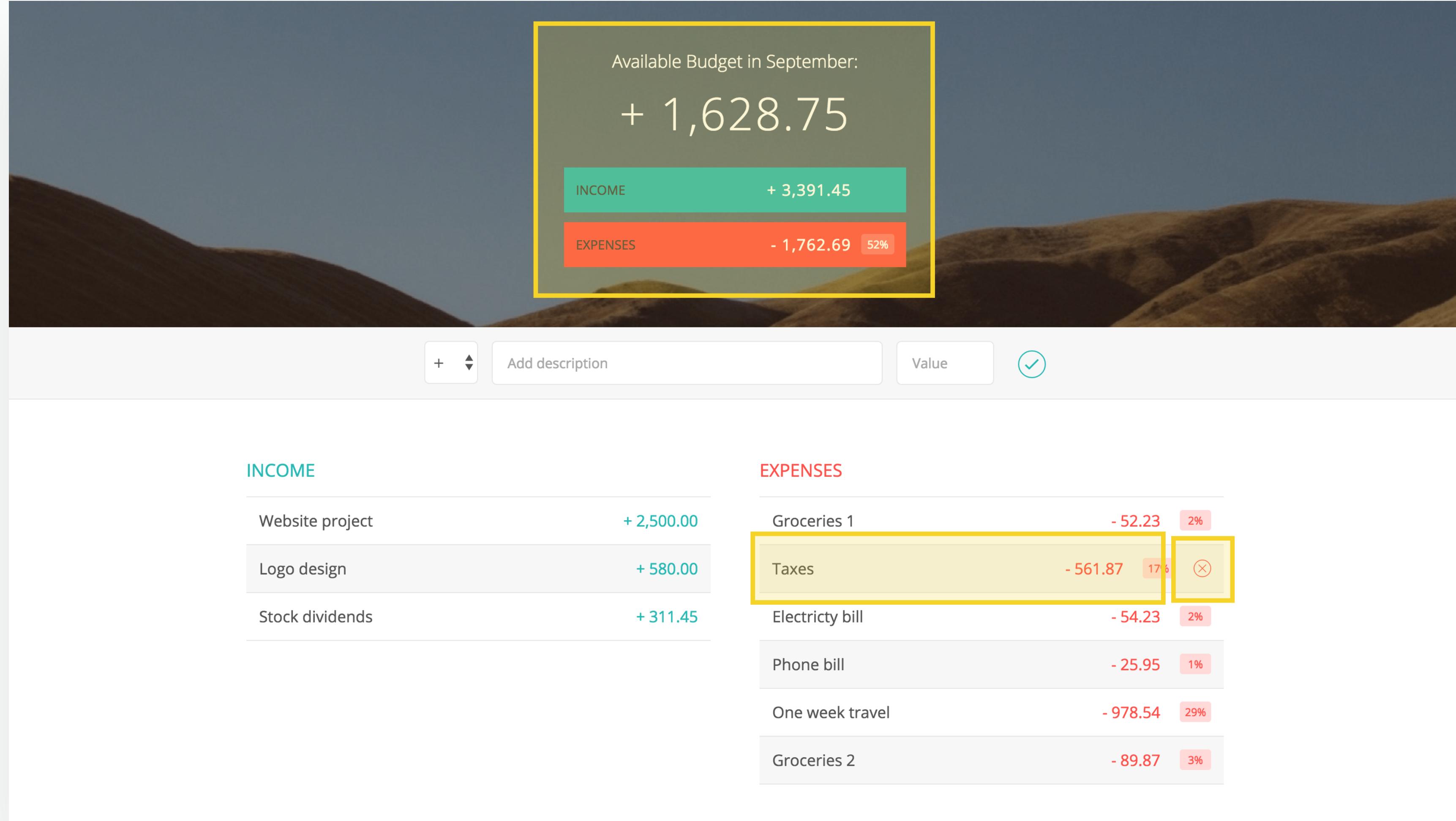
Jonas Schmedtmann


```

# AFTER STEP 1... (THAT WE JUST COMPLETED)



# PLANNING: STEP 2



## TO-DO LIST

Add event handler

Delete the item from our data structure

Delete the item to the UI

Re-calculate budget

Update the UI



codingheroes.io

```
        <td> </td>
        <td><img alt="Image Link" href="http://imgur.com/1234567" /></td>
```

## THE COMPLETE JAVASCRIPT COURSE

SECTION

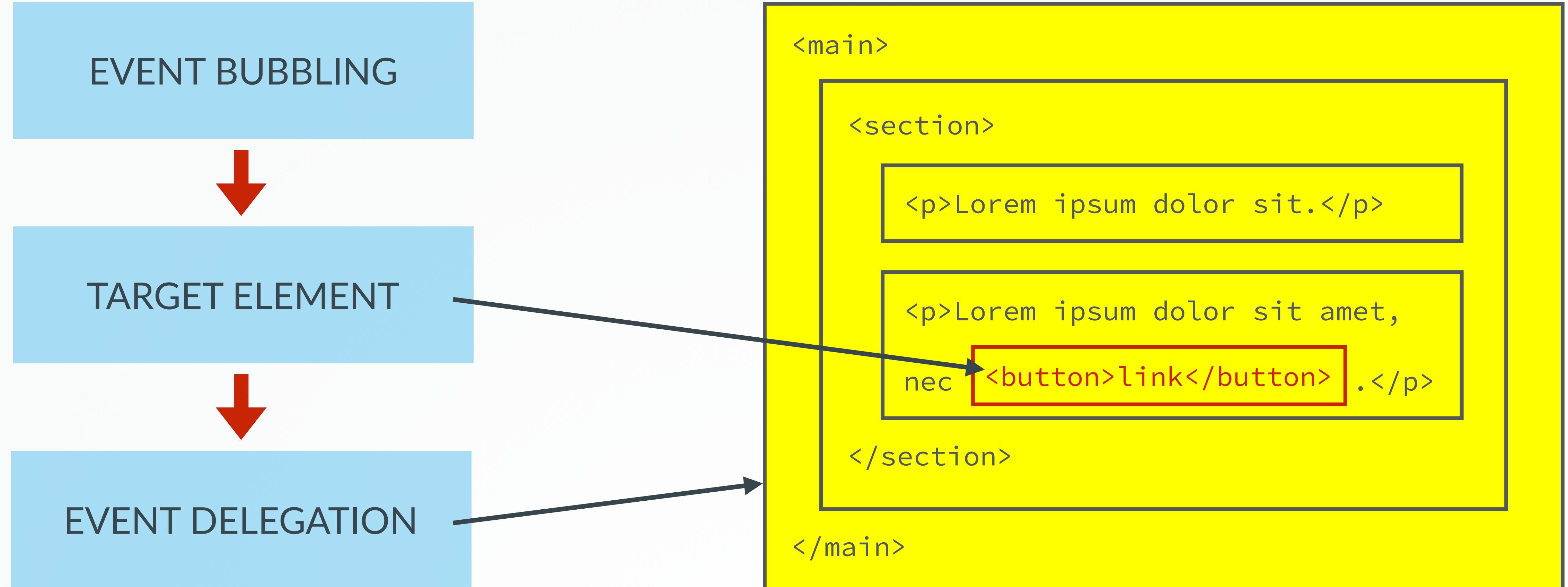
PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

LECTURE  
EVENT DELEGATION



**Jonas Schmedtmann**

# EVENT BUBBLING, TARGET ELEMENT AND EVENT DELEGATION



# WHEN TO USE EVENT DELEGATION

## USE CASES FOR EVENT DELEGATION

1. When we have an element with lots of child elements that we are interested in;
2. When we want an event handler attached to an element that is not yet in the DOM when our page is loaded.



codingheroes.io

## THE COMPLETE JAVASCRIPT COURSE

SECTION

PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

LECTURE

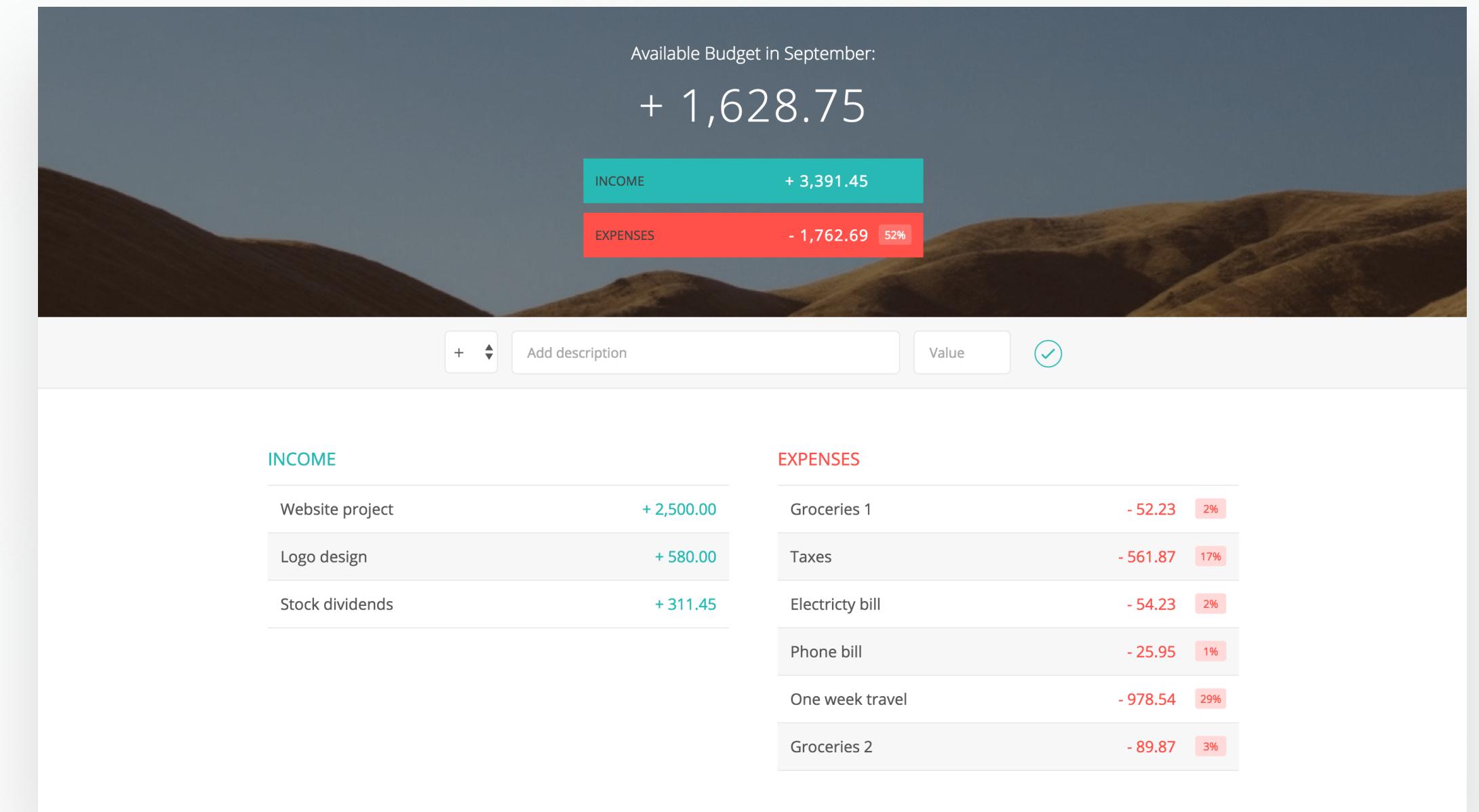
SETTING UP THE DELETE EVENT  
LISTENER USING EVENT  
DELEGATION



Jonas Schmedtmann

# WHAT YOU WILL LEARN IN THIS LECTURE

- How to use event delegation in practice;
- How to use IDs in HTML to connect the UI with the data model;
- How to use the parentNode property for DOM traversing.





codingheroes.io

## THE COMPLETE JAVASCRIPT COURSE

SECTION

PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

LECTURE

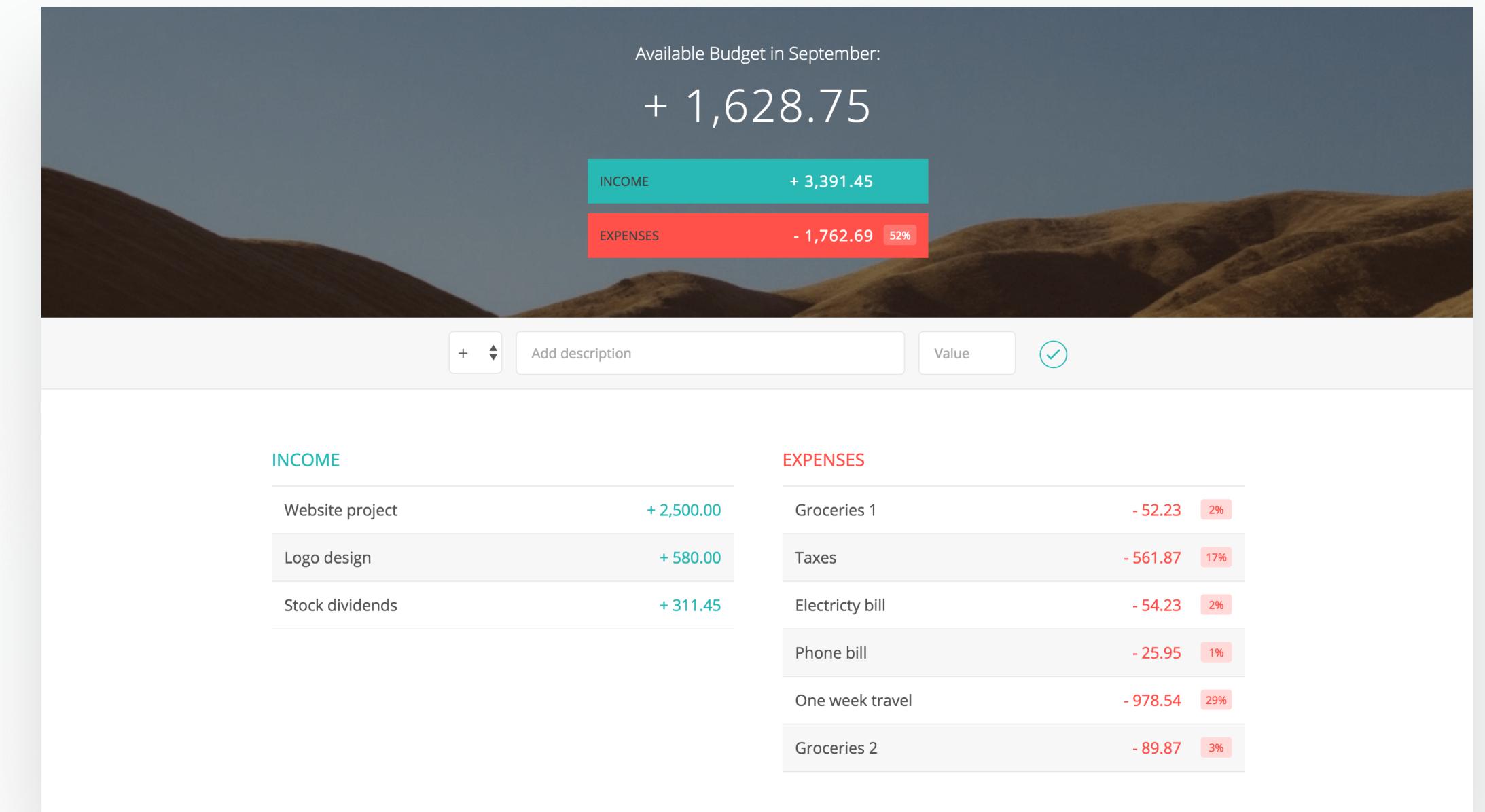
DELETING AN ITEM FROM OUR  
BUDGET CONTROLLER



Jonas Schmedtmann

# WHAT YOU WILL LEARN IN THIS LECTURE

- Yet another method to loop over an array: map;
- How to remove elements from an array using the splice method.





codingheroes.io

```
function( 'post' );  
function( 'comment' );  
function( 'category' );  
function( 'tag' );  
  
if( $row == 1 ) $columns = 0;  
else $columns = 1;  
$row++;  
  
if( $row == 1 ) $last = true;  
else $last = false;  
  
if( $row == 1 ) $image_link = true;  
else $image_link = false;  
  
if( $row == 1 ) $attachment = wp_get_attachment_image();  
else $attachment = esc_attr( implode( '  
', $attachment ) );  
$attachment = esc_attr( get_the_title() );  
  
if( $row == 1 ) $content = '

View details

';  
else $content = '

View details

';  
  
if( $row == 1 ) $content .= '

View details

';
```

## THE COMPLETE JAVASCRIPT COURSE

SECTION

PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

LECTURE

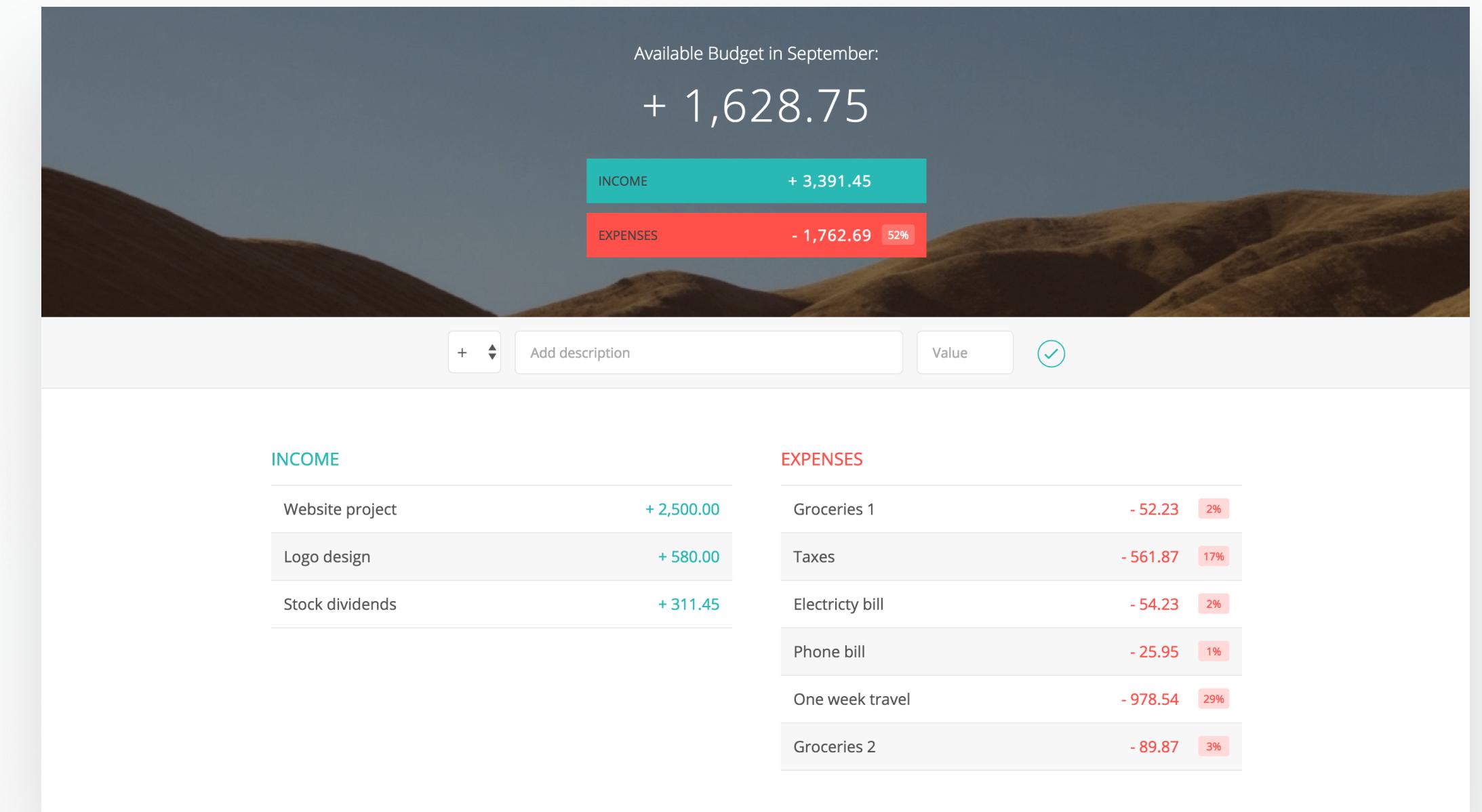
DELETING AN ITEM FROM THE UI



Jonas Schmedtmann

# WHAT YOU WILL LEARN IN THIS LECTURE

- More DOM manipulation: how to remove an element from the DOM.





**coding**heroes.io

## THE COMPLETE JAVASCRIPT COURSE

SECTION

PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

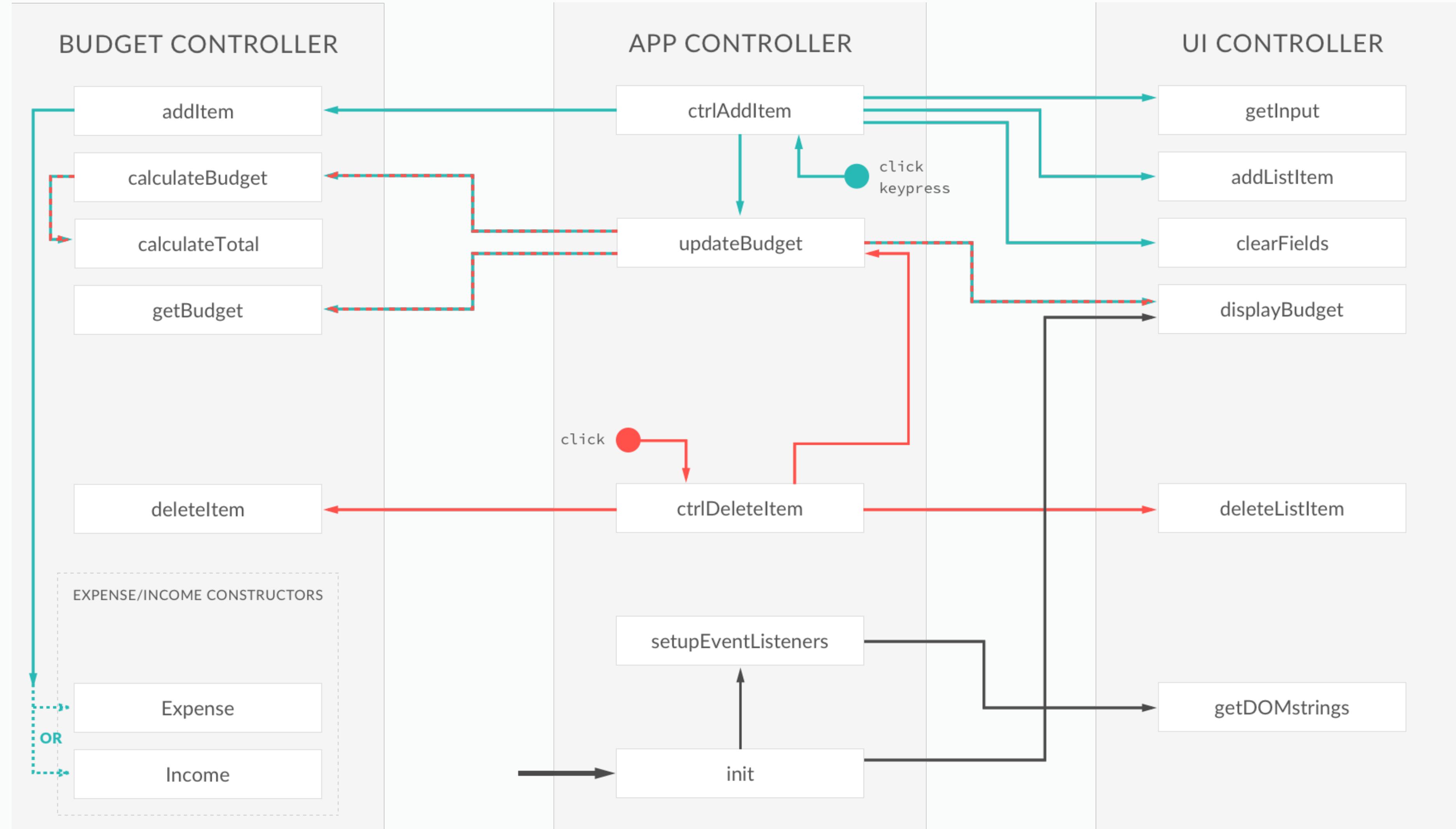
LECTURE

PROJECT PLANNING AND  
ARCHITECTURE: STEP 3

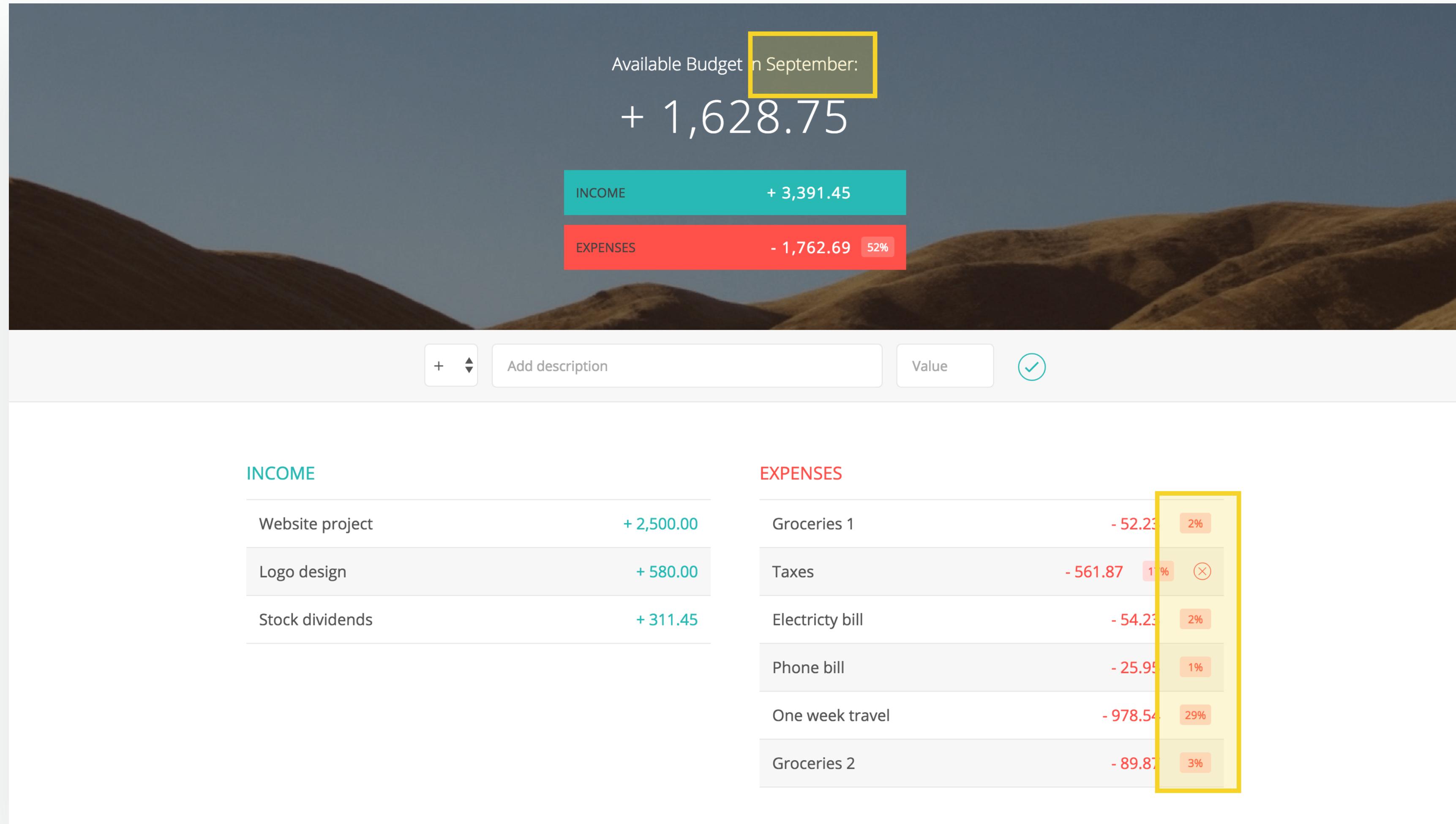


**Jonas Schmedtmann**

# AFTER STEP 2... (THAT WE JUST COMPLETED)



# PLANNING: STEP 3



## TO-DO LIST

Calculate percentages

Update percentages  
in UI

Display the current  
month and year

Number formatting

Improve input field  
UX



A circular portrait of a man with short brown hair and a light beard, smiling at the camera.

# Jonas Schmedtmann

# THE COMPLETE JAVASCRIPT CURSE

# SECTION

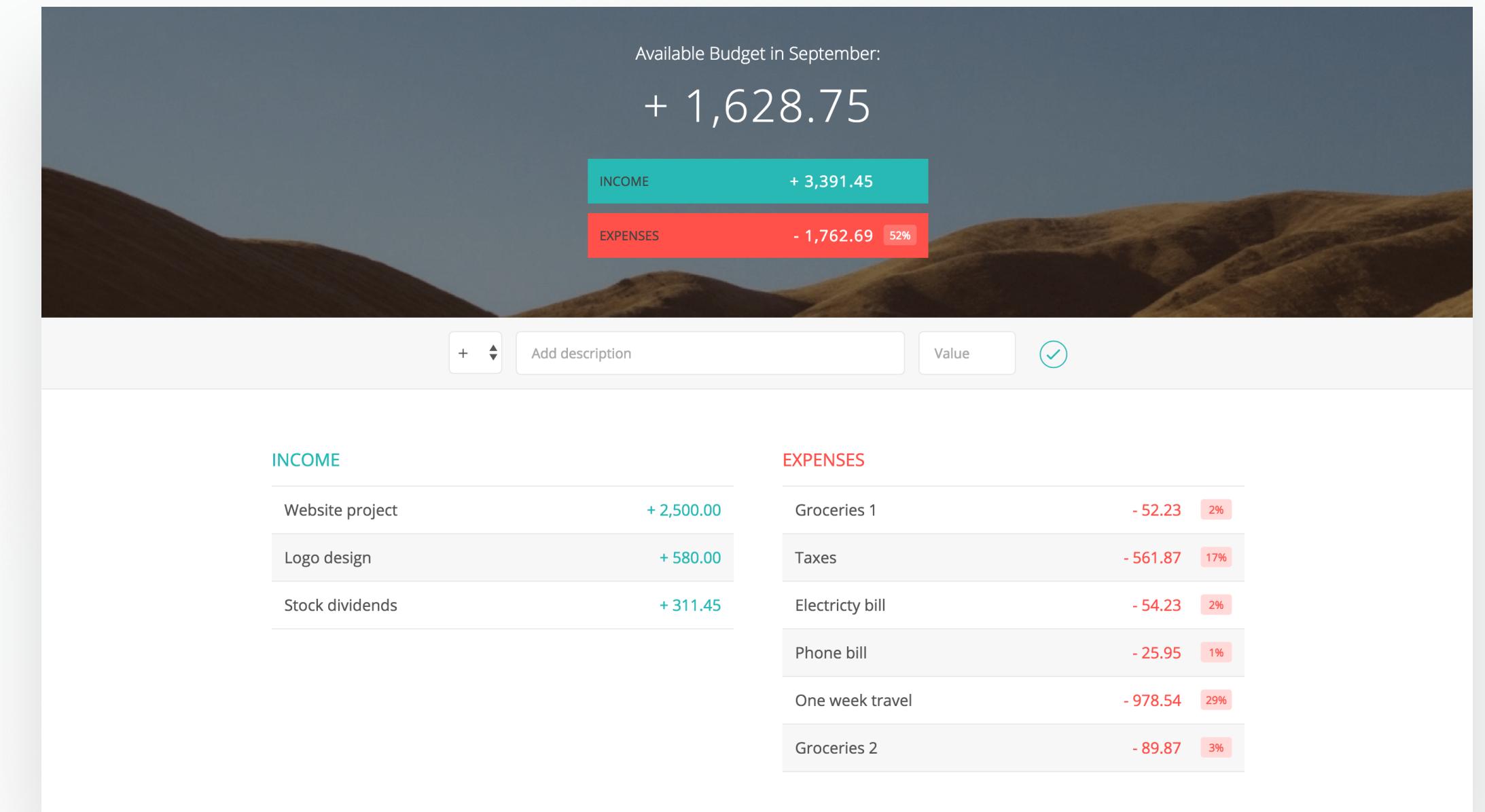
# PUTTING IT ALL TOGETHER: THE BUDGET APP PROJECT

# LECTURE

# UPDATING THE PERCENTAGES: CONTROLLER

# WHAT YOU WILL LEARN IN THIS LECTURE

- Reinforcing the concepts and techniques we have learned so far.





codingheroes.io

## THE COMPLETE JAVASCRIPT COURSE

SECTION

PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

LECTURE

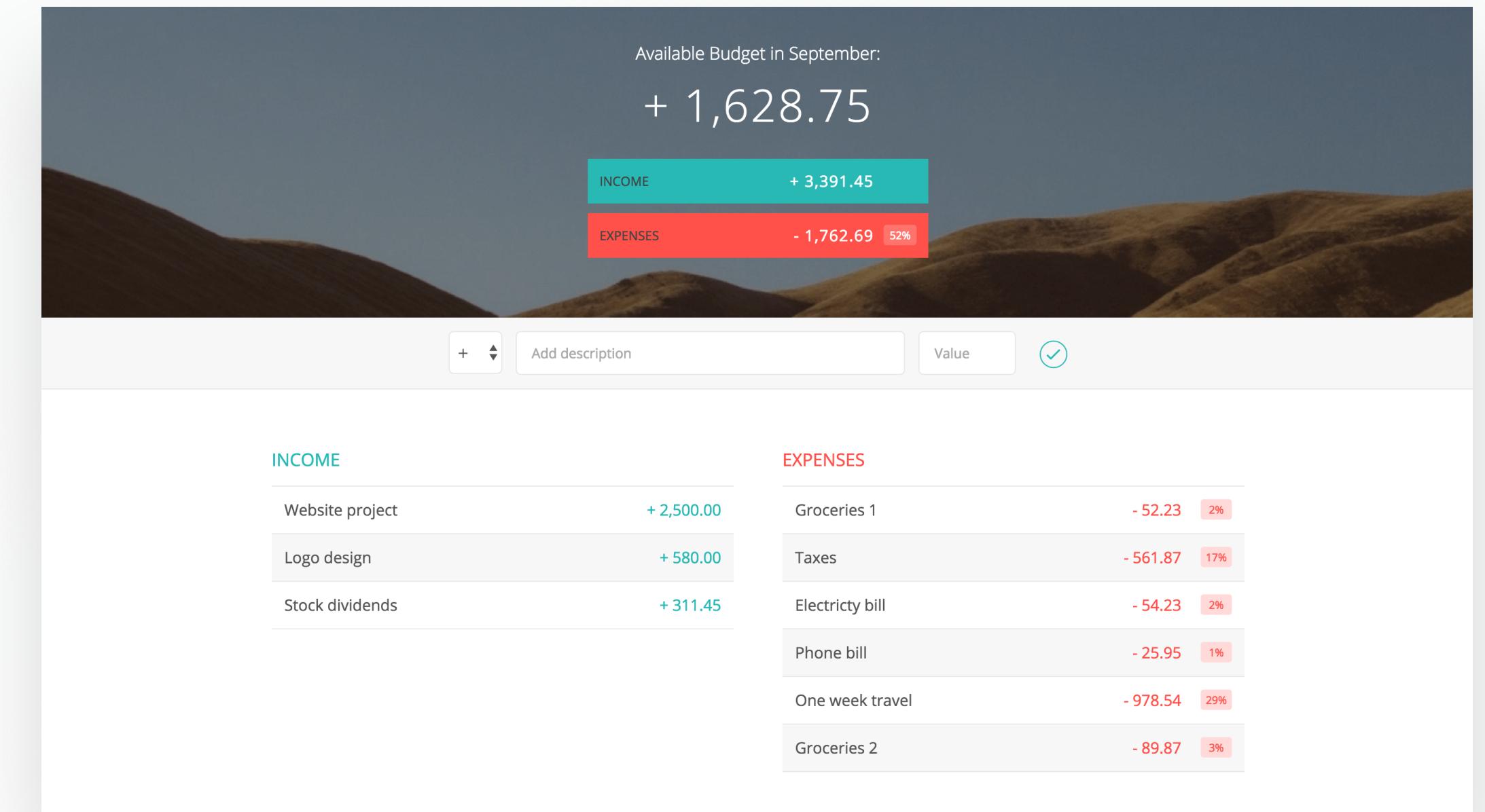
UPDATING THE PERCENTAGES:  
BUDGET CONTROLLER



Jonas Schmedtmann

# WHAT YOU WILL LEARN IN THIS LECTURE

- How to make our budget controller interact with the Expense prototype.





**coding**heroes.io

```
function( 'post' );  
  
if( !empty( $post ) ) {  
    $post = wp_get_post($post);  
  
    if( $post->post_type == 'attachment' ) {  
        $post->post_type = 'image';  
        $post->post_mime_type = 'image';  
        $post->post_content = wp_get_attachment_image_src($post->ID, 'full');  
        $post->post_excerpt = esc_attr( implode( '  
', array_map( 'wp_kses_post', $post->post_excerpt ) ) );  
        $post->post_title = esc_attr( get_the_title($post->ID) );  
        $post->post_content = wp_get_attachment_image_src($post->ID, 'full');  
        $post->post_excerpt = esc_attr( implode( '  
', array_map( 'wp_kses_post', $post->post_excerpt ) ) );  
        $post->post_title = esc_attr( get_the_title($post->ID) );  
    }  
}  
  
return $post;
```

## THE COMPLETE JAVASCRIPT COURSE

SECTION

PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

LECTURE

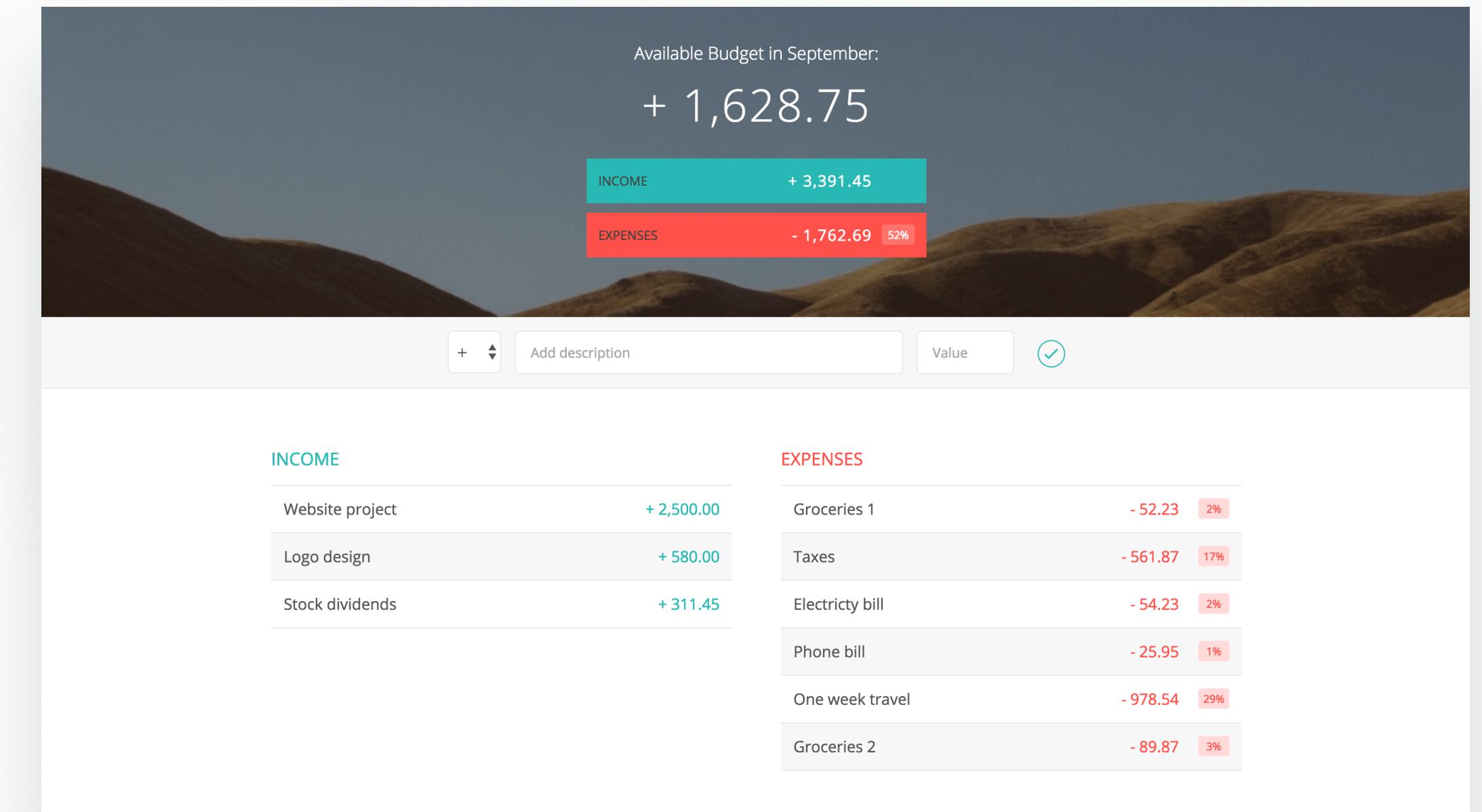
UPDATING THE PERCENTAGES: UI  
CONTROLLER



**Jonas Schmedtmann**

# WHAT YOU WILL LEARN IN THIS LECTURE

- How to create our own `forEach` function but for `nodeLists` instead of arrays.





**coding**heroes.io

## THE COMPLETE JAVASCRIPT COURSE

SECTION

PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

LECTURE

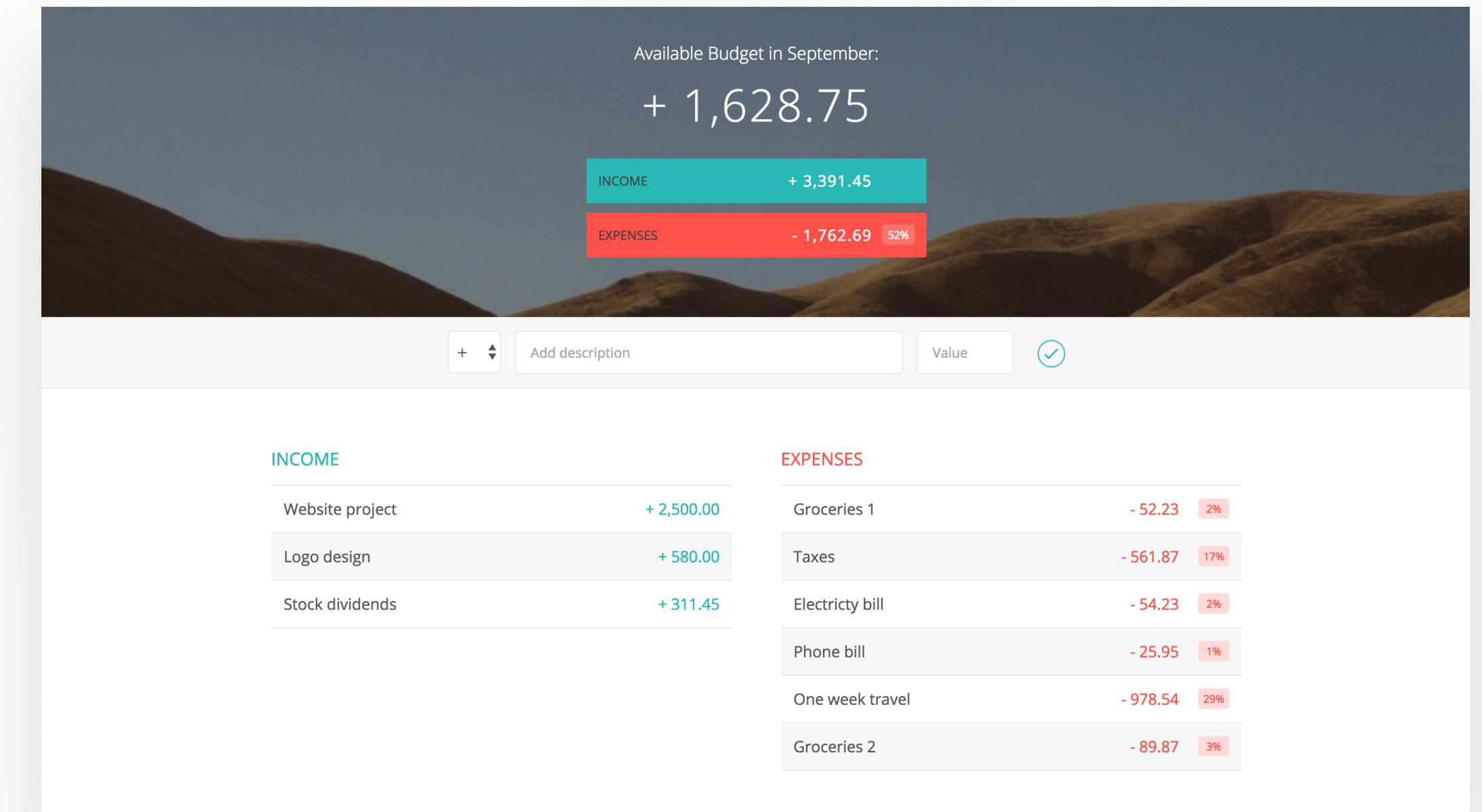
FORMATTING OUR BUDGET  
NUMBERS: STRING  
MANIPULATION



**Jonas Schmedtmann**

# WHAT YOU WILL LEARN IN THIS LECTURE

- How to use different String methods to manipulate strings.





codingheroes.io

## THE COMPLETE JAVASCRIPT COURSE

SECTION

PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

LECTURE

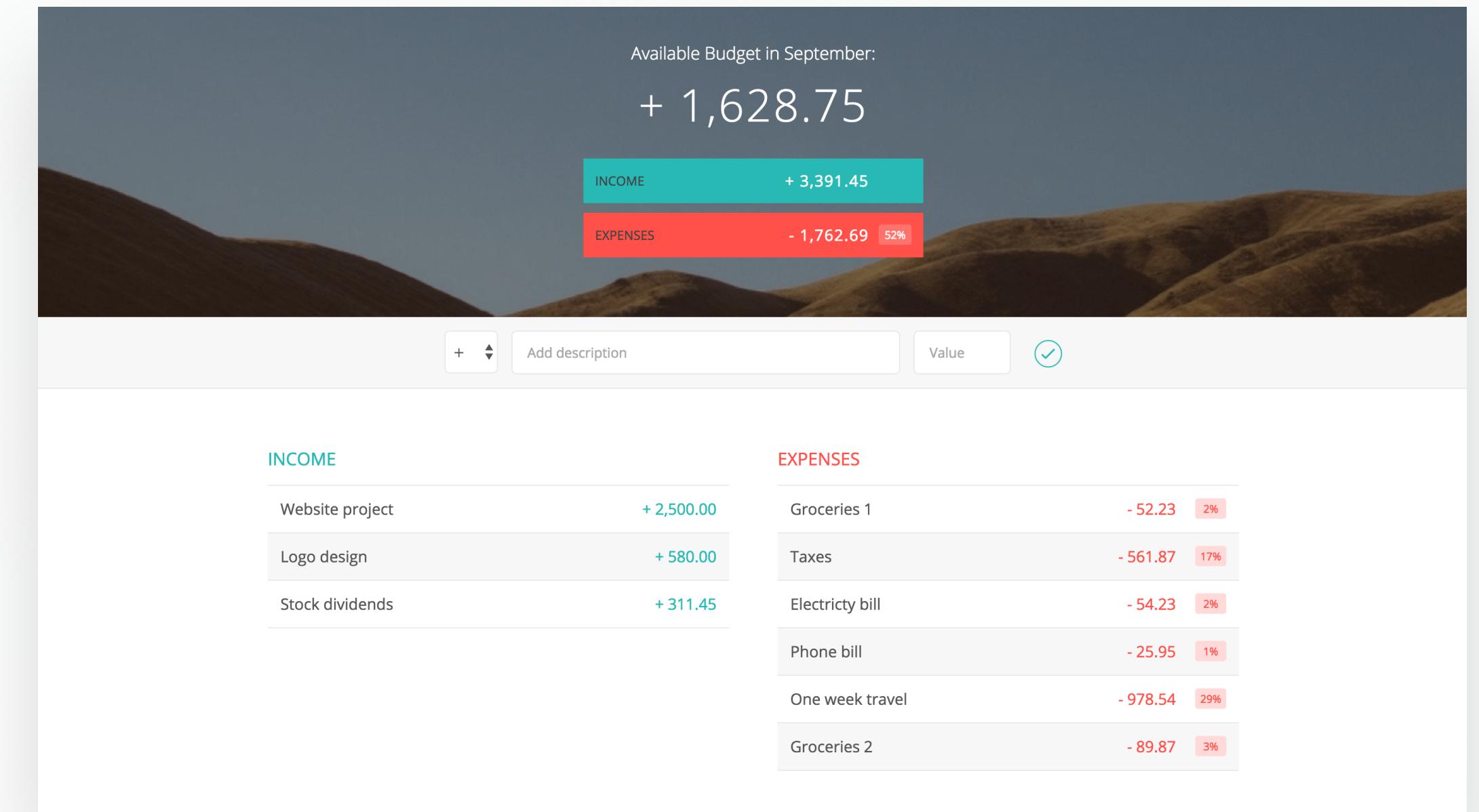
DISPLAYING THE CURRENT  
MONTH AND YEAR



**Jonas Schmedtmann**

# WHAT YOU WILL LEARN IN THIS LECTURE

- How to get the current date by using the Date object constructor.





# Jonas Schmedtmann



# THE COMPLETE JAVASCRIPT CURSE

# SECTION

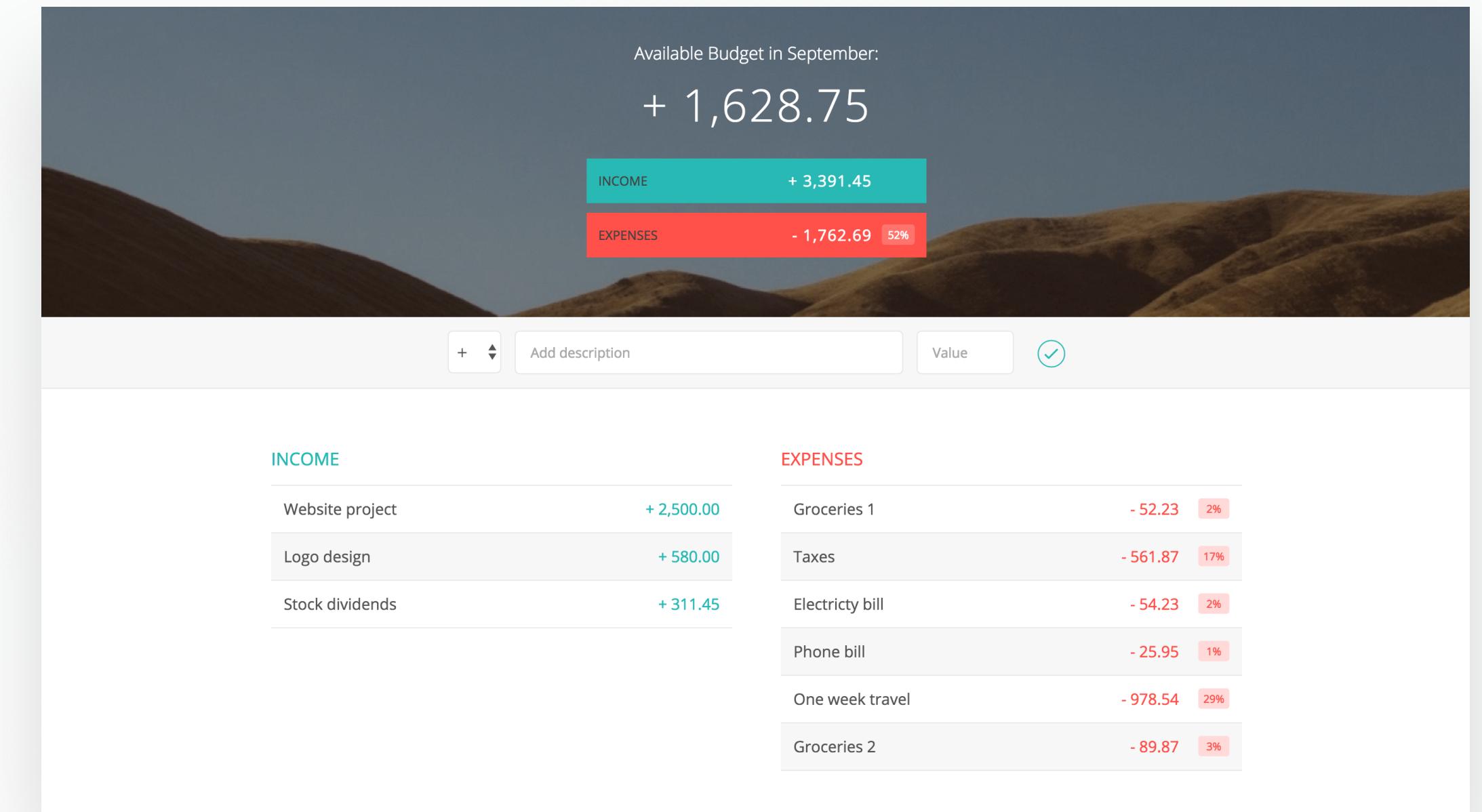
# PUTTING IT ALL TOGETHER: THE BUDGET APP PROJECT

# LECTURE

# FINISHING TOUCHES: IMPROVING THE UX

# WHAT YOU WILL LEARN IN THIS LECTURE

- How and when to use 'change' events.





codingheroes.io

## THE COMPLETE JAVASCRIPT COURSE

SECTION

PUTTING IT ALL TOGETHER: THE  
BUDGET APP PROJECT

LECTURE

WE'VE MADE IT! FINAL  
CONSIDERATIONS



Jonas Schmedtmann

# OUR FINAL ARCHITECTURE

