

Web Dev in 2019

Yesterday
was just the beginning.

```
78 // ... strlen( realpath($_SERVER['DOCUMENT_ROOT']) ) . '._CAPTCHA&t=' . u  
79 $_SESSION['_CAPTCHA']['config'] = serialize($captcha_config);  
80  
81 return array(  
82     'code' => $captcha_config['code'],  
83     'image_src' => $image_src  
84 );  
85 }  
86  
87
```

```
88 if( !function_exists('hex2rgb') )  
89     function hex2rgb($hex_str, $return_string = false, $separator = ',')  
90     {  
91         $hex_str = preg_replace("/^[0-9A-Fa-f]/", '', $hex_str);  
92         $rgb_array = array();  
93         if( strlen($hex_str) == 6 ) {  
94             $color_val = hexdec($hex_str);  
95             $rgb_array['r'] = ($color_val >> 0x10);  
96             $rgb_array['g'] = ($color_val >> 0x8);  
97             $rgb_array['b'] = ($color_val >> 0);  
98         } elseif( strlen($hex_str) == 3 ) {  
99             $rgb_array['r'] = hexdec(str_repeat(substr($hex_str, 0, 1), 2));  
100             $rgb_array['g'] = hexdec(str_repeat(substr($hex_str, 1, 1), 2));  
101             $rgb_array['b'] = hexdec(str_repeat(substr($hex_str, 2, 1), 2));  
102         } else {  
103             return false;  
104         }  
105         return $return_string ? implode($separator, $rgb_array) : $rgb_array;  
106     }  
107  
108 // Draw the image  
109 if( isset($_GET['captcha']) )  
110     {  
111         $code = $_SESSION['_CAPTCHA']['code'];  
112         $image_src = $_SESSION['_CAPTCHA']['image_src'];  
113         // ...  
114     }
```



Where to go next?

What we'll take a peek at now

- JavaScript
- More CSS
- Testing
- Build tools
- Backend

JavaScript

The web tech stack

- Content
 - HTML
- Design / Layout
 - CSS
- Behaviour / Interactivity
 - JavaScript

A teaser for the upcoming talk

```
<button>Click me!</button>
```

```
<!-- Let's make the button do something with JavaScript! -->
```

```
<script>
```

```
  document.querySelector('button').onclick = function() {  
    alert('Hello world!');  
  }
```

```
</script>
```

More CSS

```
346 .widget-area-sidebar {
347     font-size: 13px;
348 }
349
350
351 /* =Menu
352
353
354
355 #access {
356     display: inline-block;
357     height: 69px;
358     float: right;
359     margin: 11px 28px 0px 0px;
360     max-width: 800px;
361 }
362
363 #access ul {
364     font-size: 13px;
365     list-style-type: none;
366     padding: 0 0 0 -0.81em;
367     margin: 0;
368     z-index: 9999;
369     text-align: right;
370 }
371
372 #access li {
373     display: inline-block;
374     text-align: left;
```


Your CSS will probably grow.

<code>img { /*... */ }</code>	<code>/* all images... */</code>
<code>img.hero { /* ... */ }</code>	<code>/* the big landing page image */</code>
<code>img.hero2 { /* ... */ }</code>	<code>/* another big image? */</code>
<code>.hero-container { /* ... */ }</code>	<code>/* a container for big images? */</code>
<code>.herocontainer1 { /* ... */ }</code>	<code>/* I give up. HELP! */</code>

Methods to keep it tidy

- OOCSS: Object-oriented CSS
- BEM: Block-Element-Modifier
- ACSS: Atomic CSS
- SMACSS: Scalable & modular architecture

A sample for BEM

```
.product { /*...*/ }           /* a product tile */
.product__img { /*...*/ }       /* the product image */
.product__title { /*...*/ }     /* the product title */
/* modified styles for highlighted (featured) products */
.product--featured { /*...*/ }
.product__title--featured { /*...*/ }
```

Testing

A laboratory setting with a person in a white lab coat and blue gloves holding a beaker of blue liquid. In the background, there is a microscope and various glassware containing green and yellow liquids.

Automated testing

- Encourages better code design
- Faster development time, higher confidence
- Test categories
 - Unit tests
 - End-to-end tests

Test example: Unit test

```
test('adds 1 + 2 and returns 3', function() {  
  expect(add(1, 2)).toBe(3);  
});
```

```
test('adds -1 and 1 and returns 0', function() {  
  expect(add(-1, 1)).toBe(0);  
});
```

Test example end-to-end test

```
describe('Shop page', function() {  
  it('clicking "buy" goes to the checkout page', function() {  
    cy.visit('https://example.org/shop')  
    cy.contains('buy').click()  
    // Should be on a new URL which includes '/checkout'  
    cy.url().should('include', '/checkout')  
  })  
})
```

Testing tools

- Jest
- Karma
- Jasmine
- Cypress
- Ava

A detailed black and white photograph of a mechanical gear train. The image shows multiple gears of different sizes meshed together, mounted on shafts. The lighting highlights the metallic surfaces and the intricate details of the teeth and bolts. The overall composition is dense and technical.

Build tools

Automate what takes you time

- You often have to repeat certain tasks
- Examples
 - Check if your code is correct & looks right
 - Resize images
 - Run your tests
 - Minimize your code

Use build tools to do that automatically

- Webpack
- Gulp
- Grunt
- npm scripts

Backend



When you need to do more...

- Not everything can be done in the browser
- Not everything should be done in the browser
- Backends let you store data & do processing
 - Process a transaction
 - Store user data

Backend programming

- JavaScript (node.js)
- Ruby
- Python
- PHP
- Java

A note on tools



Web Dev tooling



A dense collage of various JavaScript and web development logos and icons. The logos include: npm, yarn, flow, BABEL, tc, webpack, ESLint, JS Hint, Marionette, Backbone.js, TS, Lo, Jasm, ember, 2015 ES, JUnit, jQuery, Knockout, and others. The logos are arranged in a chaotic, overlapping manner, representing the ecosystem of JavaScript frameworks and tools.

A collection of various hand tools is scattered on a dark, textured surface. The tools include a hammer with a silver head and a wooden handle, a pair of pliers with yellow and black handles, a wrench with a black handle, a screwdriver with a red handle, and several drill bits of different sizes. The text "Different tools solve different problems." is overlaid in white, bold, sans-serif font across the center of the image.

Different tools solve
different problems.

Tools...

- It's good to have a toolbox
- Use the simplest tool that solves the problem
- If a tool makes your life harder, try another
- Don't listen to the haters



Keep learning & make the web

