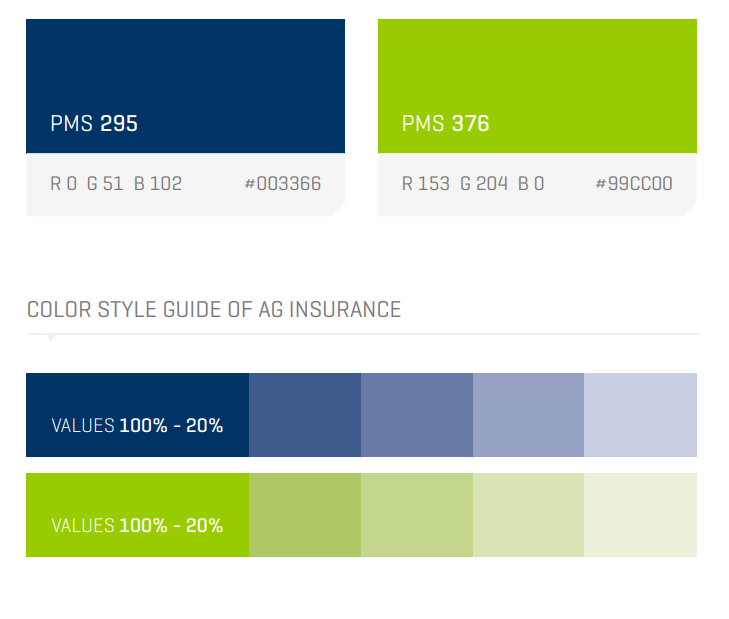
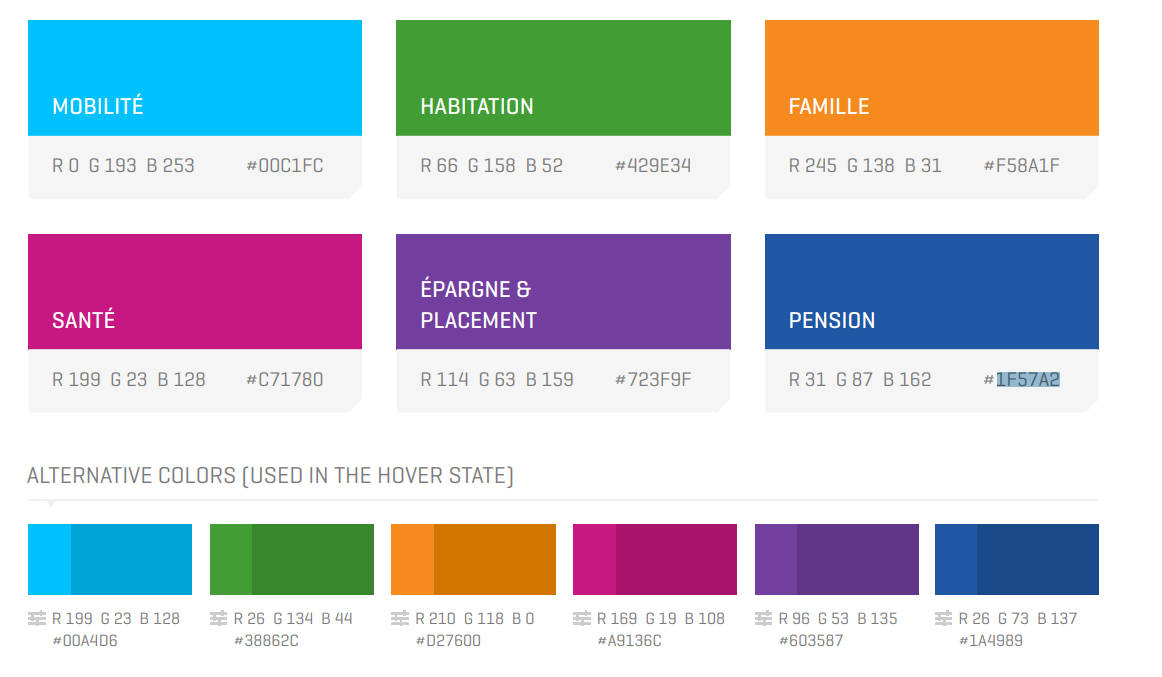
This is not (just) a button

**Distributed Front End Road Show 2023**

Build Your Own Button

# AG Style Cheat

Colors  




Fonts

* (AG) Geogrotesque, Arial, Note this

# Start with a basic HTML

Create a new .html file in your notepad (notepad (++), sublime text, visual studio code, …)

* Enter your HTML
  + A HTML file will always start with <!DOCTYPE html> to declare to the browser that what follows is HTML
  + HTML is built by using attributes these are defined by tags
    - There are container attributes that are declared by an opening <X> and closing tag </X>; these container attributes may contain other attributes
    - There are empty attributes that cannot contain other attributes, these are declared by only using an opening tag
  + HTML will mainly be structured by
    - a head element, which contains meta data about the HTML page
      * title
      * links
      * …
    - a body element that will contain the info that is being displayed in the page
      * headers <h1>, <h2>, <h3>, …
      * paragraphs <p>
      * Images
    - You can add comments to your HTML enclosed by this tag “<!—comment here -->
* Save your html

You try it:

Create a HTML page with the title “Build Your Own Button” that contains a header: “This is a heading”, a paragraph: “this is a paragraph” and a button with the text: “this is not just a button!”.[[1]](#footnote-1)



# Let’s add some style

We have our button but it’s not nice. Let’s make it more stylish.

## Basic style

* Choose the attribute you want to style, in our case it will be the button
* Add the style attribute to the begin tag and feed it the CSS properties and values

<attribute style= “property: value”>

* To chain multiple style elements a semicolon “;” as separator.

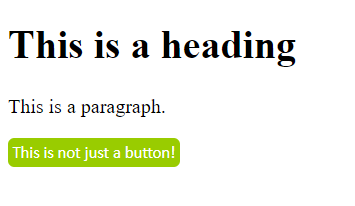
This is called inline CSS

You try it:

* Change the button as following[[2]](#footnote-2)

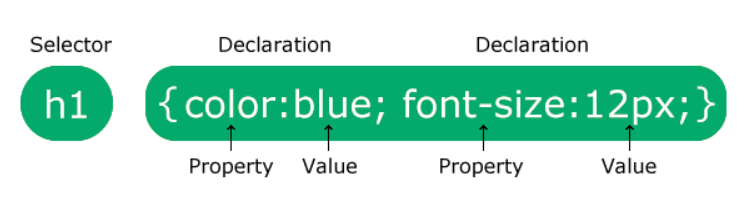
|  |  |
| --- | --- |
| **Property** | **Value** |
| * Background-color | AG green (see cheat sheet) |
| * Font-Family | Calibri |
| * (font) color | White |
| * Border-color | Transparent |
| * Border-radius | 5px |
| * Padding | 2px |

To get this result:



## Abstract style from HTML by using CSS

### CSS syntax



The selector indicates the element within your HTML you wish to style.

There are different kinds of selectors (we will only focus on simple ones):

* Simple 🡪 select elements based on name, (#)id, (.)class
* Combinator 🡪 select elements based on specific relationships between them
* Pseudo-class 🡪 select elements based on a certain state
* Pseudo-elements 🡪 select and style a part of an element
* Attribute 🡪 select the elements based on an attribute or attribute value

CSS stands for cascading style sheets: this means the style will be applied in a waterfall way: first the least specific style is applied but will be overwritten by the style elements defined by the more specific selectors.

F. ex: if we were to define a font color on the body selector, all fonts within this selector will have the same color except if they belong to more specific elements that have other font colors attributed.

Comments in CSS are indicated by /\* comment \*/. Using \* as selector will affect every html element on the page.

You try it:

* **What would you expect the output of your HTML page to be if you applied following CSS[[3]](#footnote-3)**

\* {font-family: “Arial”} /\*all elements on the page will use the Arial font \*/

Body {color: red} /\*apply red to all text that is within the body \*/

H1 {color: blue} /\* apply blue to all header texts\*/

p.AG {color: green} /\* apply green to all paragraphs of the class AG \*/

* **What would you expect if you changed you <p> element in the HTML to   
  <p class=”AG”>?[[4]](#footnote-4)**
* **What if you removed the font family property in your button tag?[[5]](#footnote-5)**

### Decoupling

There are two ways to decouple the style elements from the HTML tags both use CSS

* Internal CSS   
  🡪 centralized place where you change the style elements within the HTML file  
  To do so include the <style> element inside the head container
* External CSS  
  🡪 CSS is contained in a separate file linked in the HTML file. Changing the link to the style sheet will change the whole style of your HTML

To do so include the reference to the required style sheet

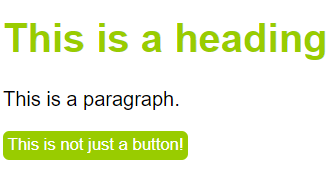
You try it:

* Move the styling from the button element in between the styling element of the head element. Check if your page stayed the same.[[6]](#footnote-6)
* Now move all between the Style element tags to a new .css file (button.css) and include the link in your HTML head element.[[7]](#footnote-7)
* Create a copy of this style sheet (button2.css) and make some changes to some style elements. Change the link for the stylesheet in your HTML to reference this style sheet. What happens?

## Sass

Sass is a CSS preprocessor which reduces repetition of CSS and therefore saves time. This is specifically useful if your site can have multiple main colors. Each time you would use the main color you would have to copy paste it a lot. Instead of doing so SASS allows you to use placeholders that will be filled in with the actual values you define in one single place.  
🡪 when you need to change it you only must change it in one place.

For example, the following SASS code

$universe: #99CC00;

\*{font-family: arial}

button {

background-color: $universe

color: white;

border-color: transparent;

border-radius: 5px;

padding: 2px

}

h1 {color:$universe}

Instead of using #99CC00 twice (once for the button background and once for the header font, we use a placeholder that will be replaced by the value we attribute to the placeholder when the sass code compiles to CSS

This would enable us to change the color scheme easily depending on the universe we would use the HTML for (same page other color scheme).

Changing one value in the above styling would move our theme from green to blue

$universe: #003366;

button {

background-color: $universe

color: white;

border-color: transparent;

border-radius: 5px;

padding: 2px

}

h1 {color:$universe}

as it would transpile to the following CSS

\* {font-family: arial}

h1 {color: #003366}

button {background-color: #003366; color: white; color: transparent;  
border-radius: 5px; padding: 2px}

# Let’s make a dynamic button

We now have a button, that is styled as we would like it but otherwise rather static.

To add some action, we’ll use JavaScript.

## JavaScript

We won’t cover the whole JavaScript here but just the basics

Like CSS there are three ways to add JavaScript:

* **Inline**: add an action (JavaScript code) linked to an event within the tag of the element for which you want this behavior

<Button Event=” JavaScript Code”>

* **Internally**: use the <script> anywhere in you document. Within this element define your code. To select the element on which the code applies the most common used method is:

document.getElementById()

* **Externally**: as is the case with CSS we will just export the code that would have been incased within the script element to a separate file with the “.js” extension and import it in the HTML referencing the source[[8]](#footnote-8)

<script src=”link to your file here”><\script>

## Bootstrap (W3.CSS)

Bootstrap is a free CSS framework that helps with the **responsive** component of CSS.  
By responsive we mean that the lay out of your HTML elements will change depending on the device and screen orientation.

It does so by applying different styles to the same elements depending on your screen.  
🡪 when defining screen elements as bootstrap classes they inherit the behavior of those classes.

To use the libraries

**W3.CSS:** <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">

**Bootstrap:**

<!-- Latest compiled and minified CSS -->  
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet">  
<!-- Latest compiled JavaScript -->  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></script>

Or download it and include it in your HTML

## A responsive button

Using JavaScript, we can change the button based on events on this button, states of this button and the screen of our device.

You try it:

* Change the color of the button when you click on it[[9]](#footnote-9)

Event: onClick

JavaScript:

function alternateUniverse() {

console.log(document.getElementById("BYOB")[[10]](#footnote-10).style);[[11]](#footnote-11)

document.getElementById("BYOB").style.backgroundColor = "#99CC00";

}

Note that this will only change the color once as we overwrite the background color and have no way to go back to the original. How could you change this?

* Change the button so the color changes when you hover over it to a 20% darker shade

Within CSS you can define styles for specific states of your element (pseudo-class selectors see CSS ). This is done by appending “:pseudo-class” to your selector.

Therefore, change your CSS file so that our specific button with id = BYOB turns darker with the on mouse over link[[12]](#footnote-12)

Note if we would have adapted the background-color attribute rather than the filter the change would only appear before we click the button and not afterwards. Why is this?

* And finally, make it so the button only contains an icon when you open the page on an extra small device.
  + As the resizing of the page influences your elements it’s good to use bootstrap for this
  + For this you will need an icon, there are multiple icon libraries you can import into your project
  + Use bootstraps responsive utilities (hidden and visible)[[13]](#footnote-13)
* Import bootstrap into your project, include jQuery as well as bootstrap relies on jQuery

<script src="http://code.jquery.com/jquery-2.1.3.min.js" data-semver="2.1.3" data-require="jquery"></script>

<!-- Latest compiled and minified CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet">

<!-- Latest compiled JavaScript -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></script>

* Import an icon library into your project
  + Bootstrap  
    <!-- Option 1: Include in HTML -->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.3.0/font/bootstrap-icons.css">

/\* Option 2: Import via CSS \*/

@import url("https://cdn.jsdelivr.net/npm/bootstrap-icons@1.3.0/font/bootstrap-icons.css");

* + Font-awesome
  + ….

You should have something like this

**CSS:**

@import url("https://cdn.jsdelivr.net/npm/bootstrap-icons@1.3.0/font/bootstrap-icons.css");

\* {font-family: arial;}

h1 {color:#003366;}

button {

background-color: #003366;

color: white;

border-color: transparent;

border-radius: 5px;

padding: 2px

}

#BYOB:hover{ filter: brightness(120%); }

**HTML:**

<!DOCTYPE html>

<html>

<head>

<title>Build your own Button</title>

<script src="http://code.jquery.com/jquery-2.1.3.min.js" data-semver="2.1.3" data-require="jquery"></script>

<!-- Latest compiled and minified CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet">

<!-- Latest compiled JavaScript -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></script>

<link rel="stylesheet" href="button.css">

</head>

<body>

<h1>This is a heading</h1>

<p class = "AG">This is a paragraph.</p>

<button id= "BYOB" onclick="alternateUniverse()">

<div class="d-none d-sm-block"> This is not just a button! </div>

<div class="d-block d-sm-none bi-alarm"></div>  
</button>

</body>

<script>

function alternateUniverse() {

console.log(document.getElementById("BYOB").style);

document.getElementById("BYOB").style.backgroundColor == "#99CC00")

</script>

</html>

Or we could just ask ChatGPT (he’s not quite there yet):

"Please create a web page that contains a button with a grass green background that turns darker on hover and changes from green to blue on click and back. The button should contain the text 'I'm not (just) a button' when displayed on any device larger than medium, but should only contain an icon of an apple when the screen becomes smaller than a medium device."



1. <!DOCTYPE html>

   <html>

   <head>

   <title>Build your own Button</title>

   </head>

   <body>

   <h1>This is a heading</h1>

   <p>This is a paragraph.</p>

   <button>This is not just a button!</button>

   </body>

   </html> [↑](#footnote-ref-1)
2. * <button style="background-color:#99CC00;Font-family:Calibri;color:white; border-color: transparent; border-radius:5px; padding:2px">

   [↑](#footnote-ref-2)
3. Blue heading, red paragraph 🡪 all in arial font, except the button which would be in Calibri font [↑](#footnote-ref-3)
4. Blue heading, green paragraph 🡪 all in arial font, except the button which would be in Calibri font [↑](#footnote-ref-4)
5. Blue heading, green paragraph 🡪 all in arial font [↑](#footnote-ref-5)
6. <head>

   <title>Build your own Button</title>

   <style>  
   \* {font-family: arial}

   button {

   background-color: #99CC00;

   color: white;

   border-color: transparent;

   border-radius: 5px;

   padding: 2px

   }

   </style>

   </head> [↑](#footnote-ref-6)
7. <head>

   <title>Build your own Button</title>

   <link rel="stylesheet" href="button.css">

   </head>

   <body> [↑](#footnote-ref-7)
8. This might be an internal path or a web URL [↑](#footnote-ref-8)
9. <button id= "BYOB" onclick="alternateUniverse()"> is not just a button! </button> [↑](#footnote-ref-9)
10. As you will need to reference the element by id add an id to your button [↑](#footnote-ref-10)
11. Use Console.log to log things in your console. You can access the console by pressing F12 when you are in the browser window. This case will show you the style elements of your button [↑](#footnote-ref-11)
12. #BYOB:hover{ filter: brightness(120%); } [↑](#footnote-ref-12)
13. Include two different div’s inside your button; one containing the text the other one containing the icon. Alternate when which is shown. [↑](#footnote-ref-13)