



# Cascading Style Sheets (CSS)



**HTML**



**JS**



**CSS**





amazon

Amazon Video

Off to College

Sponsored by BLACK + DECKER

Departments - Your Amazon.com Today's Deals Gift Cards & Registry Sell Help

Amazon Video TV Shows Movies Shorts

Available to Buy

GAME OF THRONES Season 6

All Videos Included with Prime Add-on Subscriptions Rent or Buy Free to Watch

Included with Prime

Originals & Exclusive TV

See More

WILL PUGH PRESENTS

ALL-STAR

NOTHING

OUR DEMOCRACY HAS BEEN HACKED

MR. ROBOT

PBS

THE FINAL SEASON OF DOWNTON ABBEY

HUMANS

BOARDWALK EM

Add-on Subscriptions

See All Subscriptions

Add-on Subscription

STARZ

Get STARZ—captivating original series and hit movies

Learn more

OUTLANDER

Add-on Subscription

SHOWTIME

Get SHOWTIME—award-winning Original Series, hit movies, sports, and more

Learn more

HOMELAND

Add-on Subscription

SEES0

Get Seeso—Original and classic comedy, NBC late night, stand-up and more

Learn more

Rent or Buy

Top Movies

Based on what customers are

THE WOLF OF WALL STREET

THE MONUMENTS MEN

ASTRO BLY

THE MONUMENTS MEN

DEADPOOL

Amazon Try Prime

Off to College

Sponsored by BLACK + DECKER

Amazon Video

Amazon Video

Go

Departments

[Hello, Sign in Your Account](#)[Sign in Your Account](#)[Try Prime](#)[Lists](#)[Cart](#)

[Your Amazon.com](#)[Today's Deals](#)[Gift Cards & Registry](#)[Sell](#)[Help](#)

[Amazon Video](#)[TV Shows](#)[Movies](#)[Shorts](#)[Your Watchlist](#)[Your Video Library](#)[Settings](#)[Getting Started](#)[Help](#)

AMAZON ORIGINAL

The Grand Tour

We're headed stateside.

Apply for tickets to a live taping.

Available to Buy

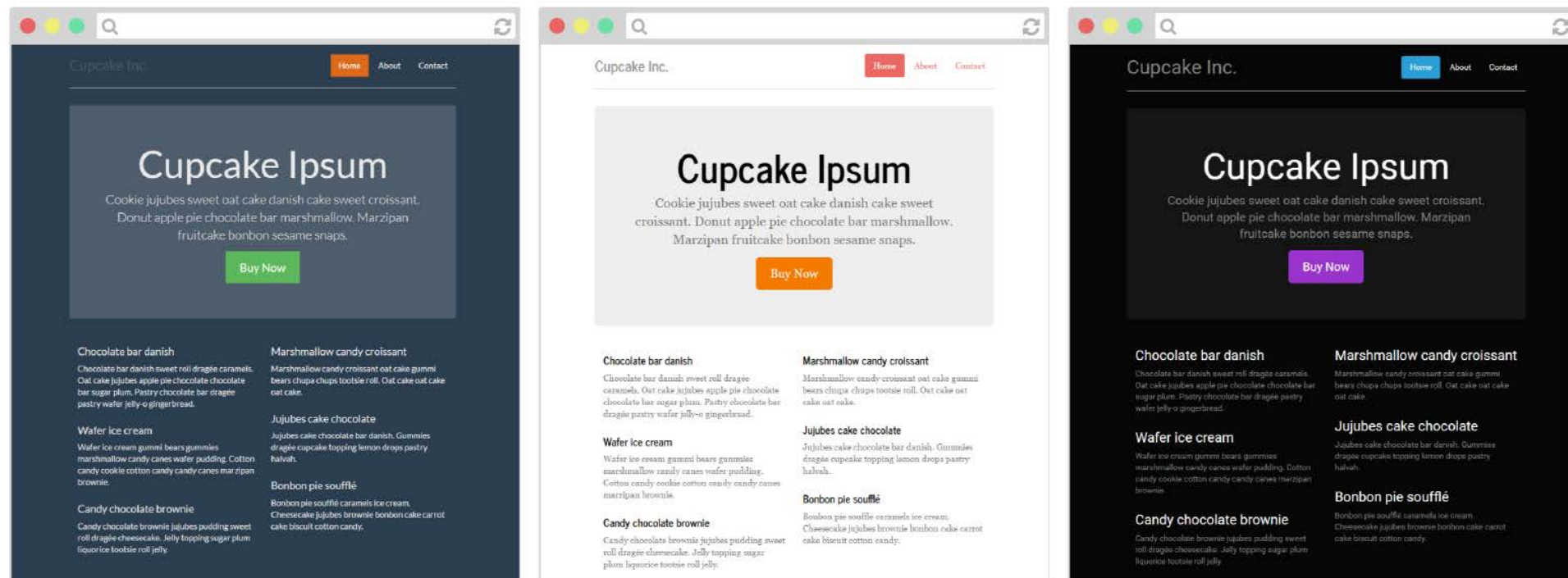
GAME OF THRONES Season 6

Add-on Subscription

STARZ

# Separation of Content & Style

- Check out: [csszengarden.com](https://csszengarden.com)



# CSS Resources

- Codecademy [HTML & CSS course](#) – free, interactive
- Shay Howe's [Learn to Code HTML & CSS](#) - written
- w3schools [CSS Tutorial](#) – semi-interactive
- [Learn CSS Layout](#) – focused on layout



# SugarCube Bleached CSS

- “Bleached” - alternate default CSS for SugarCube
- Go to [motoslave.net/sugarcube/2/](https://motoslave.net/sugarcube/2/)
- Download “Bleached [ZIP archive]”
- Copy the CSS to Story Stylesheet



# Back to HTML in VS Code



```
<!DOCTYPE html>
```

```
<html>
```

```
    <head>
```

```
        <meta charset="UTF-8">
```

```
        <title>Web Art</title>
```

```
    </head>
```

```
    <body>
```

```
        <h1>Web Art</h1>
```

```
    </body>
```

```
</html>
```




```
<head>
  <meta charset="UTF-8">
  <title>First Style</title>

  <!-- New HTML Element: style -->
  <style></style>
</head>

<body>
  <h1>Glorious Color</h1>
</body>
```



# CSS Rule



```
<head>
  <meta charset="UTF-8">
  <title>First Style</title>

  <!-- New HTML Element: style -->
  <style>
    h1 {
      color: white;
      background-color: darkviolet;
    }
  </style>
</head>

<body>
  <h1>Glorious Color</h1>
</body>
```



# CSS Rule

```
h1 {  
    color: white;  
    background-color: darkviolet;  
}
```

SELECTOR



h1 {

color: white;

background-color: darkviolet;

}




```
h1 {
```

```
  color: white;
```


```
  background-color: darkviolet;
```

```
}
```

**BLOCK**



```
h1 {  
    color: white;  
    background-color: darkviolet;  
}
```



**DECLARATION**

# Anatomy of a Declaration

The diagram illustrates the components of a CSS declaration. The text 'color: white;' is centered. Above 'color' is the label 'PROPERTY' with a white curly brace underneath it. Above 'white' is the label 'VALUE' with a white curly brace underneath it. Below the colon ':' is the label 'SEPARATOR' with a vertical white line pointing to it. Below the semicolon ';' is the label 'END' with a vertical white line pointing to it.


PROPERTY      VALUE

color: white;

SEPARATOR      END



Colors



```
color: red;  
color: #FF0000;  
color: #FF0000FF;  
color: rgb(255, 0, 0);  
color: rgba(255, 0, 0, 1);  
color: hsl(0, 100%, 50%);  
color: hsla(0, 100%, 50%, 1);
```

```
/* Keyword */  
/* RGB Hexadecimal */  
/* RGBA Hexadecimal */  
/* RGB */  
/* RGBA */  
/* HSL */  
/* HSLA */
```



# Keywords

black	dimgray	gray	darkgray	silver	lightgray	gainsboro	whitesmoke	white	rosybrown
indianred	brown	firebrick	lightcoral	maroon	darkred	red	snow	mistyrose	salmon
tomato	darksalmon	coral	orangered	lightsalmon	sienna	seashell	chocolate	saddlebrown	sandybrown
peachpuff	peru	linen	bisque	darkorange	burlywood	tan	antiquewhite	navajowhite	blanchedalmond
papayawhip	moccasin	orange	wheat	oldlace	floralwhite	darkgoldenrod	goldenrod	cornsilk	gold
khaki	lemonchiffon	palegoldenrod	darkkhaki	beige	lightgoldenrodyellow	olive	yellow	lightyellow	ivory
olivedrab	yellowgreen	darkolivegreen	greenyellow	chartreuse	lawngreen	darkseagreen	forestgreen	limegreen	lightgreen
palegreen	darkgreen	green	lime	honeydew	seagreen	mediumseagreen	springgreen	mintcream	mediumspringgreen
mediumaquamarine	aquamarine	turquoise	lightseagreen	mediumturquoise	darkslategray	paleturquoise	teal	darkcyan	aqua
cyan	lightcyan	azure	darkturquoise	cadetblue	powderblue	lightblue	deepskyblue	skyblue	lightskyblue
steelblue	aliceblue	dodgerblue	slategray	lightslategray	lightsteelblue	cornflowerblue	royalblue	midnightblue	lavender
navy	darkblue	mediumblue	blue	ghostwhite	slateblue	darkslateblue	mediumslateblue	mediumpurple	blueviolet
indigo	darkorchid	darkviolet	mediumorchid	thistle	plum	violet	purple	darkmagenta	magenta
fuchsia	orchid	mediumvioletred	deeppink	hotpink	lavenderblush	palevioletred	crimson	pink	lightpink



# Hexadecimal

- Format: #RRGGBB
- Hex: 0 – 9 then A – F

#970515

Red Green Blue

Source: [Smashing Magazine](#)

# RGB & RGBA

- Red, Green, Blue
  - Each channel goes from 0 to 255
  - Must be whole numbers
- Alpha from 0 to 1
  - 0 is transparent
  - 1 is opaque

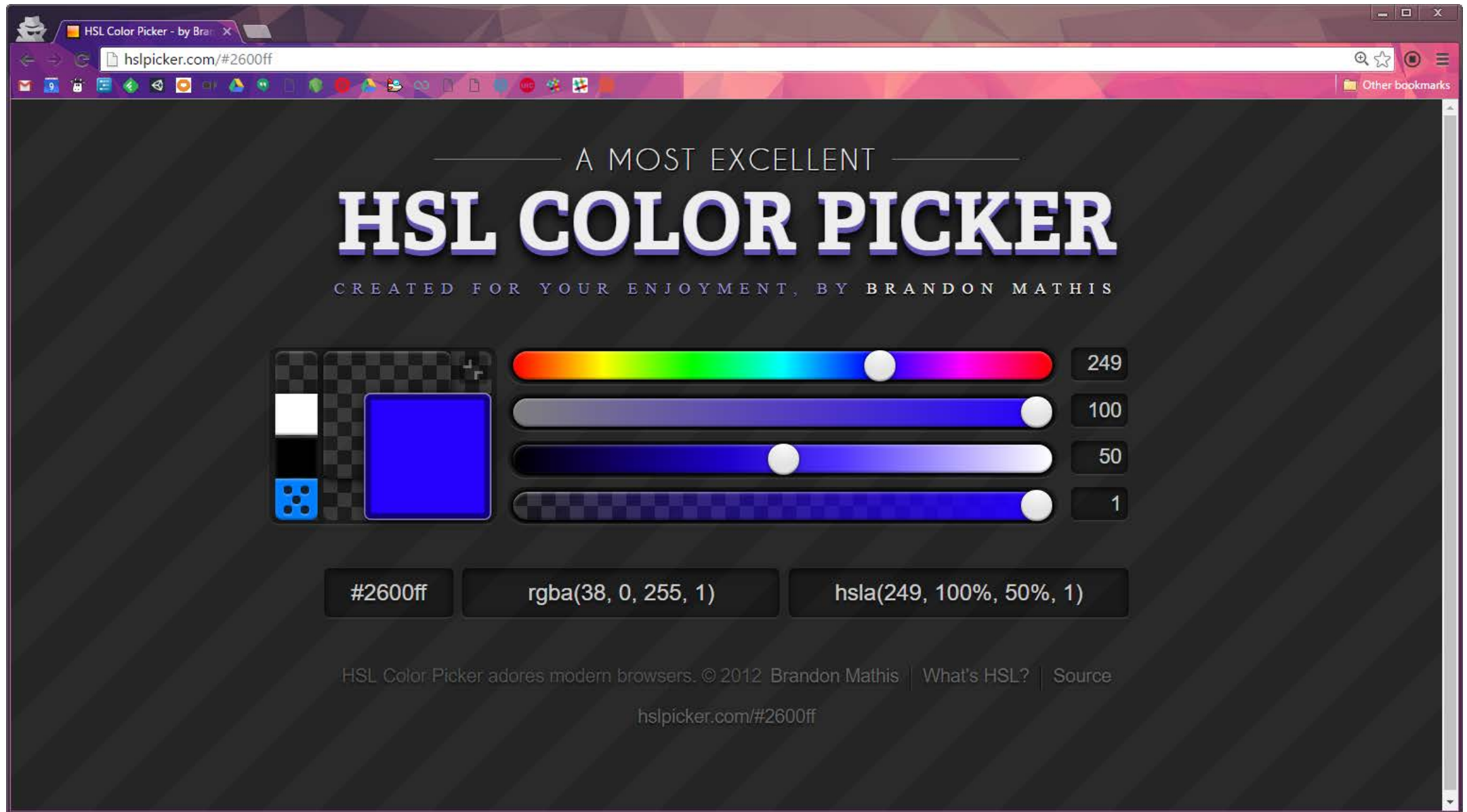
```
color: rgb(255, 101, 239);  
color: rgba(255, 101, 239, 0.5);
```



# HSL & HSLA

- Hue, Saturation, Lightness
  - Hue: 0 - 360
  - Saturation: 0 – 100 %
  - Lightness: 0 – 100 %
- Alpha from 0 to 1

```
color: hsl(187, 100%, 61%);  
color: hsla(187, 100%, 61%, 0.75);
```



<http://hslpicker.com/>

Inline Style





# Inline Style

```
<h1 style="background-color: crimson; color: white;">Inline styling. Gross.</h1>
```

# Inline Style in Twine


Type	Syntax	Example	Rendered As
Highlight, Inline	<code>@@Text@@</code>	<code>@@Text@@</code>	<code>&lt;span class="marked"&gt;Text&lt;/span&gt;</code>
→ Custom Style, Inline	<code>@@style-list<sup>[1]</sup>;Text@@</code>	<code>@@#foo;.bar;Text@@</code>	<code>&lt;span id="foo" class="bar"&gt;Text&lt;/span&gt;</code>
		<code>@@color:red;Text@@</code>	<code>&lt;span style="color:red"&gt;Text&lt;/span&gt;</code>
Highlight, Block	<code>@@ Text @@</code>	<code>@@ Text @@</code>	<code>&lt;div class="marked"&gt;Text&lt;/div&gt;</code>
→ Custom Style, Block	<code>@@style-list<sup>[1]</sup>; Text @@</code>	<code>@@#foo;.bar; Text @@</code>	<code>&lt;div id="foo" class="bar"&gt;Text&lt;/div&gt;</code>
		<code>@@color:red; Text @@</code>	<code>&lt;div style="color:red"&gt;Text&lt;/div&gt;</code>





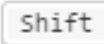




# Inspector & DevTools

# Accessing DevTools

To access the DevTools, on any web page or app in Google Chrome:



- Open the **Chrome menu**  at the top-right of your browser window, then select **Tools > Developer Tools**.
- Right-click on any page element and select **Inspect Element**.

On the keyboard:

Access DevTools	On Windows	On Mac
Open Developer Tools	 ,  +  + 	 +  + 



# Glorious Chrome

Back	Alt+Left Arrow
Forward	Alt+Right Arrow
Reload	Ctrl+R
<hr/>	
Save as...	Ctrl+S
Print...	Ctrl+P
Cast...	
Translate to English	
<hr/>	
 Adobe Acrobat	▶
 LastPass	▶
<hr/>	
View page source	Ctrl+U
Inspect	Ctrl+Shift+I

First Style

file:///E:/Dropbox/Work/SAIC/WebArt/Lessons/Day03/Day03\_Solutions/02\_FirstStyle/index.html

Elements

Console

Sources

Network

Timeline

Profiles

Application

Security

Audits

Author Settings

Styles

Computed

Event Listeners

Filter

:hov

.cls

element.style {

}

h1 {

color: #fff;

background-color: rgb(148, 0, 211);

}

h1 {

display: block;

font-size: 2em;

-webkit-margin-before: 0.67em;

-webkit-margin-after: 0.67em;

-webkit-margin-start: 0px;

-webkit-margin-end: 0px;

font-weight: bold;

}

margin 21.440

border -

padding -

380 x 37

html

body

h1

Console

top

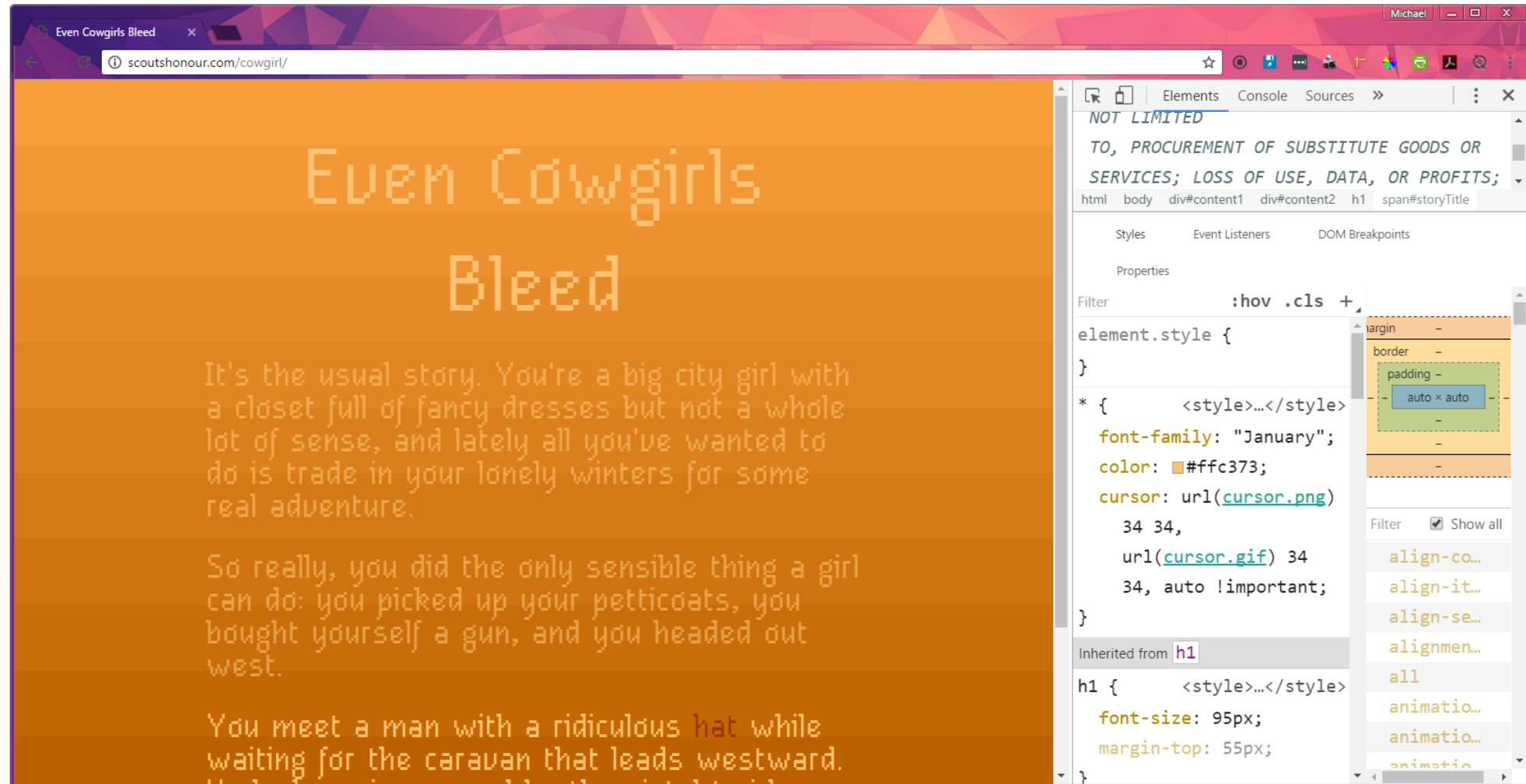
Preserve log

>

# Glorious Color

h1 380 x 37

# Inspecting a Twine





# Typography



# Font Size

- Most common units: pixels (px), percent (%), em and rem
- We'll be focusing on pixels to start

```
font-size: 35px;
```

# Fonts

## SERIF

Serif fonts have extra details on the end of the main strokes of the letters.

### EXAMPLES:

Georgia

Times

Times New Roman

## SANS-SERIF

Sans-serif fonts have straight ends to letters and therefore have a much cleaner design.

### EXAMPLES:

Arial

Verdana

Helvetica

# Fonts

## MONOSPACE

Every letter in a monospace typeface is the same width. (Non-monospace fonts have different widths.)

EXAMPLES:

Courier

Courier New

## CURSIVE

Cursive fonts either have joining strokes or other cursive characteristics, such as handwriting styles.

EXAMPLES:

Comic Sans MS

*Monotype Corsiva*

## FANTASY

Fantasy fonts are usually decorative fonts and are often used for titles. They're not designed for long bodies of text.

EXAMPLES:

**Impact**

**Haettenschweiler**



# Font Family

- Specifies which font to use
  - Every computer has different fonts installed
- Can use a generic family
  - serif, sans-serif, monospace, cursive, fantasy
- Can use a font that is stored on visitor's computer
  - See fonts on your machine: [workmark.it](http://workmark.it)

```
font-family: "Georgia";  
font-family: sans-serif;
```



# Font Stacks

- Different operating systems have different fonts
- Safe stacks: [cssfontstack.com](https://cssfontstack.com)

```
font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif;
```

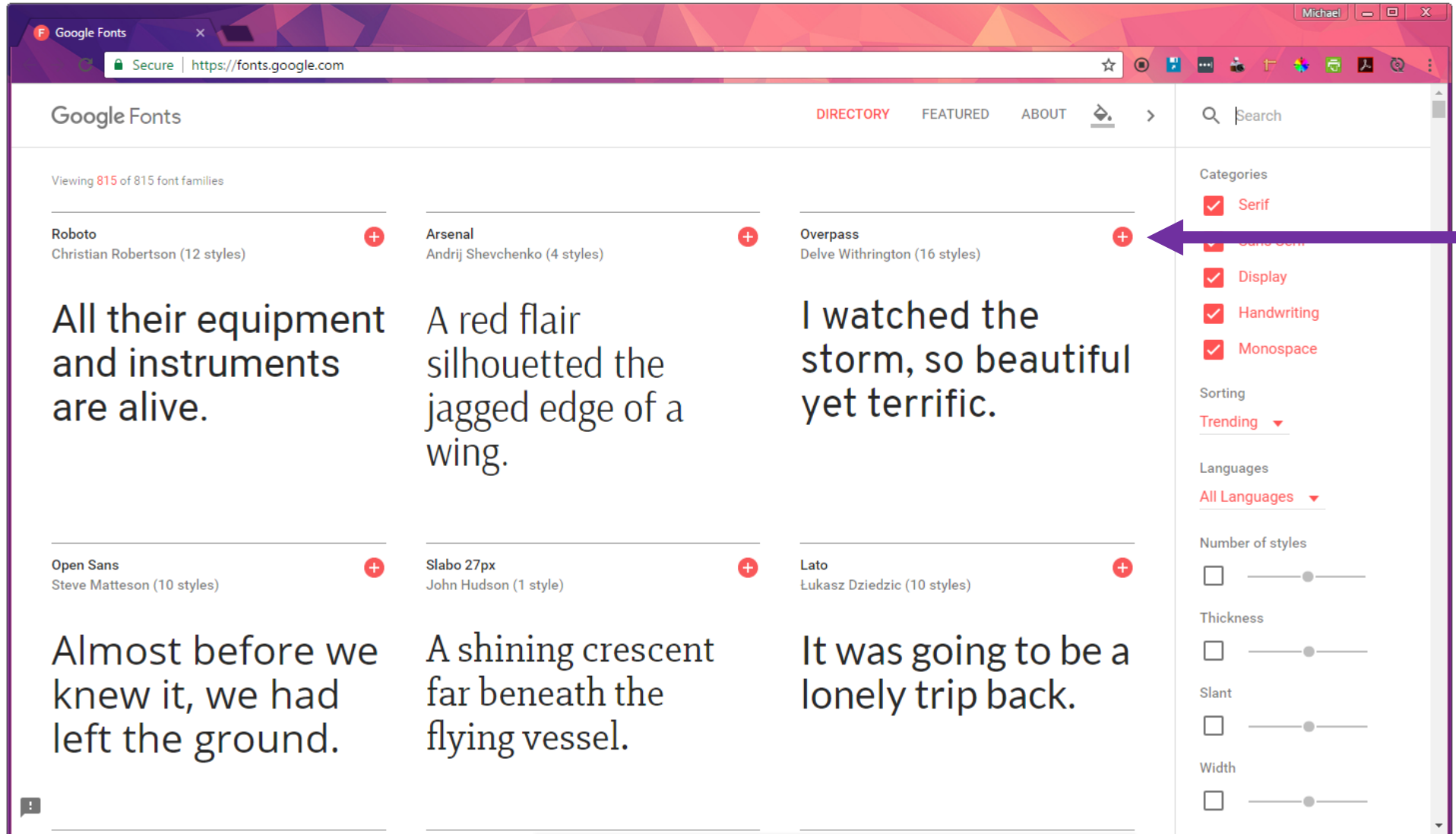


# Embedding Fonts the Google Way

# Embedding Fonts

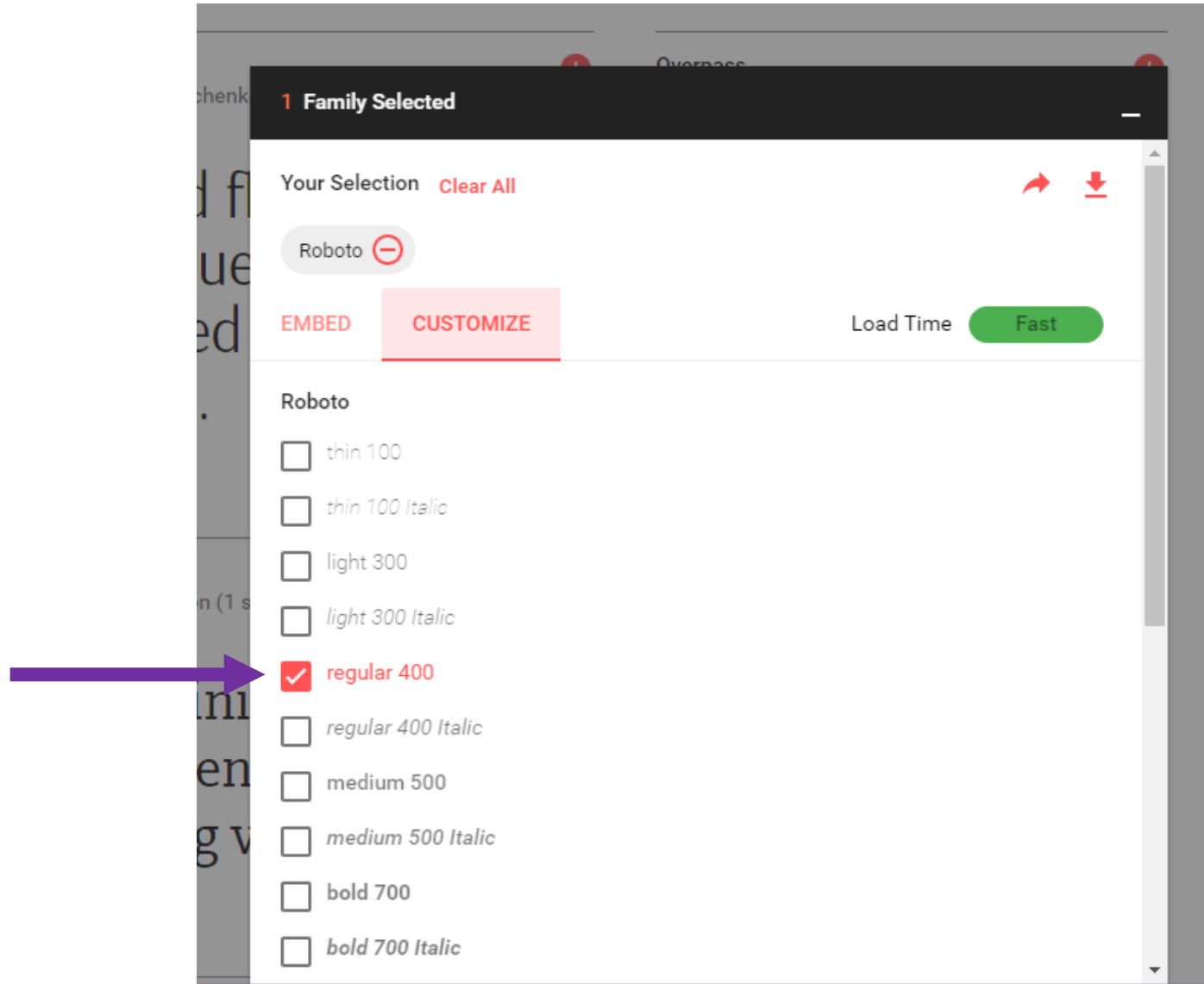
- Font files loaded over the internet
- Uses @font-face (font file support)
- Web font services: fonts.google.com, Adobe Typekit

# Pick a Font





# Pick Your Typefaces



# Get the Code

1 Family Selected

Your Selection [Clear All](#)

Roboto

EMBED CUSTOMIZE Load Time Fast

Embed Font

To embed your selected fonts into a webpage, copy this code into the <head> of your HTML document.

STANDARD @IMPORT

```
<style>
@import url('https://fonts.googleapis.com/css?family=Roboto');
</style>
```

Specify in CSS

Use the following CSS rules to specify these families:

```
font-family: 'Roboto', sans-serif;
```

For examples of how fonts can be added to webpages, see the [getting started guide](#).

```
@import url('https://fonts.googleapis.com/css?family=Roboto');
```

- @import lets you import an external CSS file from a URL
- Google Fonts has CSS files that embed a font
- Not the optimal way to use Google Fonts, but it's the best way to use it with Twine

# Typography (Continued)



# Font Weight

- Can be a number (100, 200, 300, ..., 900)
- Can be a keyword: normal (400), bold (700)
- Not all fonts are available in all weights

Raleway, 100

Raleway, 200

Raleway, 300

Raleway, 400

Raleway, 500

Raleway, 600

**Raleway, 700**

**Raleway, 800**

**Raleway, 900**

```
font-weight: 100;
```

# Text Alignment

## Left

Jelly beans tootsie roll carrot cake bonbon donut  
cotton candy lollipop powder sesame snaps. Candy  
canes gingerbread sugar plum cheesecake chupa  
chups jelly beans sesame snaps chocolate  
cheesecake. Croissant halvah powder tart donut.

## Right

Croissant cotton candy candy canes pastry. Cookie  
candy jujubes chocolate cake chocolate. Macaroon  
danish lollipop. Chocolate cake powder sugar plum  
pastry macaroon.

## Center

Liquorice gingerbread sweet toffee marshmallow  
croissant sweet roll lollipop. Dragée candy pudding.  
Topping cheesecake cake sugar plum powder  
tootsie roll. Sesame snaps liquorice cookie.

## Justify

Soufflé dessert lemon drops lollipop powder.  
Lollipop dessert ice cream bear claw danish chupa  
chups pie gummi bears jelly-o. Halvah lemon drops  
dessert. Topping powder gummies toffee powder  
apple pie chocolate bar candy tart.

```
/*  
    Possible values:  
    justify, left, right, center  
*/  
text-align: center;
```

# Font Style

- Can be a keyword: normal, italic, oblique
  - Italic – more like cursive
  - Oblique – normal font, but slanted
- If font doesn't have italic, browser will try to make it

```
font-style: italic;
```



# Line Height

- [Leading](#)
- Space between lines
- Best practice: unit-less number, based on font size

```
line-height: 1.4;
```



# Text References

- Other properties
  - text-transform
  - text-decoration
  - letter-spacing
  - word-spacing
- MDN's [Text Styling Fundamentals](#)
- w3schools [text](#) and [font](#) pages

# Typographic Tools

- [gridlover](#)
- [typescale](#)



# Getting Selective



# Element Selector

```
h1 {  
    font-size: 50px;  
}  
h2 {  
    font-size: 35px;  
}  
p {  
    font-size: 25px;  
}
```

# Combining Selectors

```
h1, h2, h3, h4, h5, h6 {  
    font-family: "Raleway", sans-serif;  
}
```



# ID Selector

- Targets 1x *unique* element
- Valid IDs:
  - Must have at least 1 character
  - No spaces
  - Best to stick to letters and dashes

```
<style>
  #title {
    font-size: 40px;
    font-style: italic;
  }
</style>
```

```
<body>
  <h1 id="title">Artist Statement</h1>
</body>
```



# Class Selector

- Targets *multiple* elements
- Same rules for valid id names apply to class names

```
<style>
  .completed {
    text-decoration: line-through;
  }
</style>
```

```
<body>
  <h1>To Do</h1>
  <ol>
    <li class="completed">Wake up</li>
    <li class="completed">Eat</li>
    <li>Run</li>
    <li>Sleep</li>
  </ol>
</body>
```



SELECTOR	MEANING	EXAMPLE
 UNIVERSAL SELECTOR	Applies to all elements in the document	<code>* {}</code> Targets all elements on the page
TYPE SELECTOR	Matches element names	<code>h1, h2, h3 {}</code> Targets the <code>&lt;h1&gt;</code> , <code>&lt;h2&gt;</code> and <code>&lt;h3&gt;</code> elements
CLASS SELECTOR	Matches an element whose <code>class</code> attribute has a value that matches the one specified after the period (or full stop) symbol	<code>.note {}</code> Targets any element whose <code>class</code> attribute has a value of <code>note</code> <code>p.note {}</code> Targets only <code>&lt;p&gt;</code> elements whose <code>class</code> attribute has a value of <code>note</code>
ID SELECTOR	Matches an element whose <code>id</code> attribute has a value that matches the one specified after the pound or hash symbol	<code>#introduction {}</code> Targets the element whose <code>id</code> attribute has a value of <code>introduction</code>
 CHILD SELECTOR	Matches an element that is a direct child of another	<code>li&gt;a {}</code> Targets any <code>&lt;a&gt;</code> elements that are children of an <code>&lt;li&gt;</code> element (but not other <code>&lt;a&gt;</code> elements in the page)
 DESCENDANT SELECTOR	Matches an element that is a descendent of another specified element (not just a direct child of that element)	<code>p a {}</code> Targets any <code>&lt;a&gt;</code> elements that sit inside a <code>&lt;p&gt;</code> element, even if there are other elements nested between them
ADJACENT SIBLING SELECTOR	Matches an element that is the next sibling of another	<code>h1+p {}</code> Targets the first <code>&lt;p&gt;</code> element after any <code>&lt;h1&gt;</code> element (but not other <code>&lt;p&gt;</code> elements)
GENERAL SIBLING SELECTOR	Matches an element that is a sibling of another, although it does not have to be the directly preceding element	<code>h1~p {}</code> If you had two <code>&lt;p&gt;</code> elements that are siblings of an <code>&lt;h1&gt;</code> element, this rule would apply to both



# Selectors Reference

- w3schools [selectors page](#)

# CSS in Twine

Edit Story JavaScript

Edit Story Stylesheet

Change Story Format

Rename Story

Snap to Grid

Story Statistics

View Proofing Copy

Published Last saved at Sat, Jan 21, 2017 7:12 PM

First Passage

Hello, welcome to my first Twine passage.

## Stylesheet



Any CSS entered here will override the default appearance of your story.

```
body {  
  font-size: 20px;  
}
```



# SugarCube HTML Structure

```
<body class="...">

  <!-- Loading screen -->
  <div id="init-screen"></div>

  <!-- Screen overlay (used with modal box in order to dim screen) -->
  <div id="ui-overlay" class="ui-close"></div>

  <!-- Container for modal box (e.g. pop up save and restart menus) -->
  <div id="ui-dialog"></div>

  <!-- Container for the sidebar -->
  <div id="ui-bar"></div>

  <!-- Container for the story -->
  <div id="story" role="main">
    <div id="passages">
      <div class="passage ..." id="..." data-passage="...">
        <!-- The active passage content -->
      </div>
    </div>
  </div>

</body>
```



# Passage IDs and Classes

- Each passage gets an ID from its name
- Each passage gets a class for each of its tags
- Body gets a class for each tag on the active passage

## General Notes

All internally created IDs and class names are normalized (which entails: removing characters which are not alphanumerics, underscores, hyphens, en-/em-dashes, or whitespace, then replacing any remaining non-alphanumeric characters with hyphens, one per group, and finally converting the result to lowercase; e.g. the passage `Gone fishin'` becomes `passage-gone-fishin`).

When displaying a passage, its tags are added to the active passage's container element and the page's `<body>` element as CSS classes. The special tags (noted below) are excluded from this mapping.

**Special Tags for Twine 1:** `debug`, `nobr`, `passage`, `script`, `stylesheet`, `widget`, and any tag starting with `twine`.

**Special Tags for Twine 2:** `debug`, `nobr`, `passage`, `widget`, and any tag starting with `twine`.

I'm the 1st



+ Tag

red ✕

big ✕

```
<!--
```

```
  This passage has an ID of "im-the-first"
```

```
  This passage has the following classes: "red", "big"
```

```
  The body element also has those classes.
```

```
-->
```

## Stylesheet



Any CSS entered here will override the default appearance of your story.

```
body.red {  
  background-color: red;  
}  
body.blue {  
  background-color: blue;  
}|
```