

15일 (목)

1. Codecademy CSS - Developing with CSS

VOD 1. Intro to Chrome Devtools

크롬 개발자도구는 CSS 에서 변화를 시도해볼 때 유용하다! 코드를 바꾸지 않고 시험삼아 바꿔볼 수 있다는 것이 장점

2. Codecademy CSS - Semantic CSS

Article 1. What Makes for a Semantic Class Name?

<https://css-tricks.com/semantic-class-names/>

보기 쉽게 클래스명을 지정해라 다른 사람이 봐도 이해할 수 있고 내가 나중에 봐도 이해할 수 있는 클래스명을 지정해야한다.

Article 2. Semantic CSS

시멘틱 css 를 사용하여 얻을 수 있는 이점을 배워보자.

이름을 짓는데 시간을 들이는 것은 아주 중요하다. css 를 유지보수하기 위해 가장 중요한 양상은 바로 Naming 이다.

Semantic vs non-semantic

```
<!-- non semantic -->
<div class="red pull-left pb3">
<div class="grid row">
<div class="col-xs-4">

<!-- semantic -->
<div class="basket">
<div class="product">
<div class="searchResults">
```

시맨틱 css 의 이점

- ⇒ 가독성 증가
 - ⇒ 반응형 사이트를 만들기 쉽다.
 - ⇒ 검색하기 쉽다.
 - ⇒ 회귀의 가능성을 줄여준다.
 - ⇒ 자동화시스템을 위한 고리를 제공한다.
- 자동화테스트는 요소와 함께 인터렉팅하고 검색되기때문이다.
- ⇒ 자바스크립트를 위한 고리를 제공한다.
 - ⇒ 디버깅하기 쉽다.
 - ⇒ html파일 사이즈를 줄인다.

3. Codecademy CSS - Learn CSS: Colors

- **Introduction to Color**

이름으로 색을 제공하기도 하고, 숫자로도 색을 제공하기도 한다.

css 에서는 세 가지 방식으로 색을 제공한다.

⇒ Named colors

⇒ RGB(red, green, blue)

⇒ HSL(hue(색조), saturation(포화), lightness(밝기))

- **Foreground color vs Background color**

```
h1 {  
  color: red;  
  background-color: blue;  
}
```

color 는 Foreground

- **Hexadecimal (16진수 컬러)**

' #' 로 시작하며, 그 후 3개 또는 6개의 문자가 들어간다. 각각 문자는 red, blue, green 을 의미한다.

```
darkseagreen: #8FBC8F
sienna:       #A0522D
saddlebrown:  #8B4513
brown:        #A52A2A
black:        #000000 or #000
white:        #FFFFFF or #FFF
aqua:         #00FFFF or #0FF
```

알파벳은 대문자나 소문자나 상관없다. 만약 위처럼 3개 로 줄여질 수 있는 사례는 6개가 2개의 쌍으로 3개로 모두 이루어져있으면 가능하다.

- **RGB Colors**

```
h1 {
  color: rgb(23, 45, 23);
}
```

숫자는 0부터 255 까지 가능 각 숫자를 16진수로 바꾼것이 #hexadecimal 컬러이다.

- **Hue, Saturation, and Lightness (HSL)**

```
color: hsl(120, 60%, 70%);
```

Hue is the first number. It refers to an angle on a color wheel. Red is 0 degrees, Green is 120 degrees, Blue is 240 degrees, and then back to Red at 360. You can see an example of a color wheel below.



Saturation 은 색조를 나타냄 0~100%

Lightness 는 밝기를 나타냄 0~100%

- **Opacity and Alpha**

```
color: hsla(34, 100%, 50%, 0.1);
```

alpha 를 추가 0~1까지 사이의 값

```
color: rgba(234, 45, 98, 0.33);
```

hex 에서도 사용이 가능하다. 00~FF 사이의 값을 가진다. (거의 안씀)

```
color: transparent;
```

color 와 같이 쓰이며, rgba(0, 0, 0, 0) 과 같은 값이다.

4. Codecademy CSS - Learn CSS: Typography (텍스트)

- **intro**

⇒ 폰트를 스타일하는법과 변경하는법

⇒ 페이지에 텍스트를 놓는 방법

⇒ 외부 폰트를 불러오는 방법 을 배울 것이다.

- **Font Family**

font-family 에 들어가는 값들의 의미를 알아보자.

⇒ **Multi-Word Values**

```
h1 {  
    font-family: 'Times New Roman';  
}
```

여러개의 단어는 콤마(',')를 써서 그룹지어주자.

⇒ **Web Safe Fonts**

윈도우나 맥에서 깨지지 않는 폰트를 찾을 수 있는 사이트이다.

<https://www.cssfontstack.com/>

⇒ **Fallback Fonts and Font Stacks**


```
h1 {  
  font-family: Caslon, Georgia,  
  'Times New Roman';  
}
```

위 예시가 폰트 스택이다. 만약 Caslon 이 브라우저에서 읽혀지지않는다면, Georgia 가 읽힌다. 만약 Georgia 도 불러오지 못한다면 마지막으로 'Times New Roman' 폰트를 불러온다. 여기서 폴백 폰트는 Caslon 을 제외한 모든 폰트이다.

⇒ Serif and Sans-Serif

SERIF

Serif fonts have extra details on the ends of the main strokes of the letters. These strokes are called serifs.

im

Serif fonts are traditionally used in print designs, as they have been considered easier to read in long paragraphs of text

SANS SERIF

Sans serif fonts lack those extra strokes on the ends of letters and have flat ends. This gives them a cleaner, more modern look.

im

Since screens have a lower resolution than print, sans serifs can look better and be easier to read.

폰트 스택은 비슷한 폰트끼리 넣어야 한다. 여기서 비슷한 폰트라는 기준이 궁금할 것이다. 바로 serif 폰트냐, sans-serif 폰트냐로 나뉜다. serif 는 글자의 디테일에 흘림체가 있는 것이

고 sans 는 흘림없이 깔끔한 글씨체인 것이다. serif 와 sans-serif 는 보통 폰트 스택의 마지막 최후의 폰트로 쓰인다.

```
h1 {  
  font-family: Caslon, Georgia,  
  'Times New Roman', serif;  
}
```

- **Font Weight**

볼드 또는 얇은 글씨체를 나타내는 프로퍼티이다. 키워드 또는 숫자를 밸류로 갖는다.

1. 키워드 밸류

⇒ bold, normal, lighter, bolder

2. 숫자 밸류

⇒ 1부터 1000 사이 값 보통 100씩 올린다. 400은 normal 700은 bold 이다.

```
.left-section {  
  font-weight: 700;  
}  
  
.right-section {  
  font-weight: bold;  
}
```

- Font Style

```
h3 {  
  font-style: italic;  
}
```

이탈릭체를 위한 것 디폴트값은 normal이다.

- Text Transformation

문자를 대문자 또는 소문자로 바꾸는 프로퍼티

```
h1 {  
  text-transform: uppercase;  
}
```

소문자는 lowercase

- **Text Layout**

텍스트의 컨테이너를 조절하는 방법을 배워보자

⇒ **Letter Spacing**

```
p {  
  letter-spacing: 2px;  
}
```

글자 하나하나의 수평적 공간을 조절하는 프로퍼티이다.

보통 가독성을 위해 사용한다. px, em 을 사용

TYPOGRAPHY

0.3em 을 준 효과

T Y P O G R A P H Y

⇒ Word Spacing

```
h1 {  
  word-spacing: 0.3em;  
}
```

단어의 수평적 공간을 조절하는 프로퍼티

이것도 가독성을 위해 사용 em 사용을 추천한다.

While typography has been practiced for many centuries, digital font design is a relatively new discipline. There are some great examples of old-school fonts (also called typefaces) which have been adapted for use on digital displays. However, I would like to highlight a few of my favorite fonts that were created with screens in mind.

0.25em 준 효과

While typography has been practiced for many centuries, digital font design is a relatively new discipline. There are some great examples of old-school fonts (also called typefaces) which have been adapted for use on digital displays. However, I would like to highlight a few of my favorite fonts that were created with screens in mind.

⇒ Line Height

The fastest cat
can race at 75

leading
font size

line height

```
p {  
  line-height: 1.4;  
}
```

라인 사이의 공간을 조절할 수 있다. 숫자만 쓰는 걸 추천한다 왜냐하면 높이에 따라 폰트 사이즈를 자동적으로 조절하기 때문이다.

While typography has been practiced for many centuries, digital font design is a relatively new discipline. There are some great examples of old-school fonts (also called typefaces) which have been adapted for use on digital displays. However, I would like to highlight a few of my favorite fonts that were created with screens in mind.

⇒ Text Alignment

```
h1 {  
  text-align: right;  
}
```

전에 배운 내용으로 텍스트의 위치를 변경한다.

TYPOGRAPHY

While typography has been practiced for many centuries, digital font design is a relatively new discipline. There are some great examples of old-school fonts (also called typefaces) which have been adapted for use on digital displays. However, I would like to highlight a few of my favorite fonts that were created with screens in mind.

TYPOGRAPHY

While typography has been practiced for many centuries, digital font design is a relatively new discipline. There are some great examples of old-school fonts (also called typefaces) which have been adapted for use on digital displays. However, I would like to highlight a few of my favorite fonts that were created with screens in mind.

text-align : justify 효과이다.

- **Web Fonts**

Web Safe Fonts

Web Fonts

Arial	BANGERS	Chewy	CODYSTAR
Verdana	CREEPSTER	Lobster	<i>Dancing Script</i>
Tahoma	Indie Flower	Pacifico	PERMANENT MARKER
Trebuchet MS	Shadows Into Light	Cutive Mono	Source Code Pro
Times New Roman	Space Mono	VT323	Montserrat Alternates
Georgia	Orbitron	Oswald	Poppins
Courier New	Raleway	Fauna One	Kiwi Maru
<i>Brush Script MT</i>	Merriweather	New Tegomin	SPECTRAL SC

web safe fonts 뿐만 아니라 셀 수 없는 web font 들이 있다.

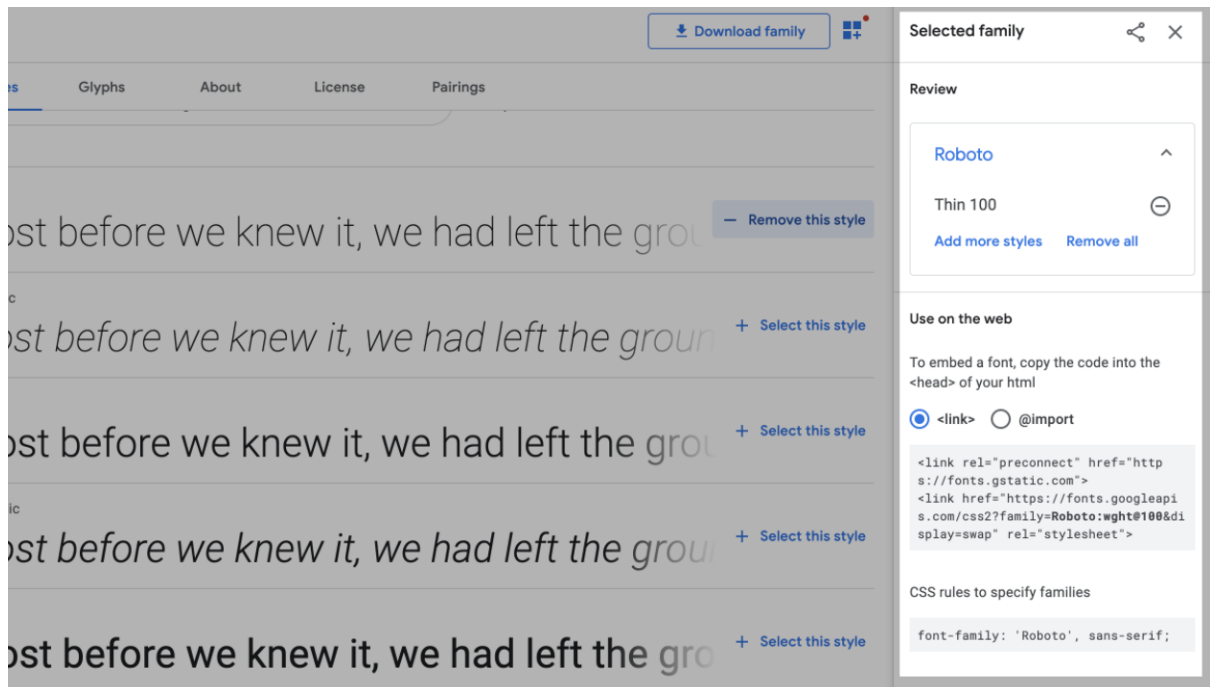
Google Fonts 나 Adobe Fonts 에서 많은 폰트를 볼 수 있다.

⇒ <https://fonts.adobe.com/> , <https://fonts.google.com/>

html 에서 link 를 이용하여 사용가능하다.

- **Web Fonts Using <link>**

웹 폰트 사이트에 들어가서 원하는 폰트를 선택 후



link 의 url 을 복사한다.

```
=swap" rel="stylesheet">
```

CSS rules to specify families

```
font-family: 'Space Mono', monospac  
e;
```

링크 아래 사용방법을 꼭 확인 후에 사용하자.

```
<head>  
  <!-- Add the link element for Google Fonts along with other metadata -->  
  <link href="https://fonts.googleapis.com/css2?  
family=Roboto:wght@100&display=swap" rel="stylesheet">  
</head>
```

후에 html의 head 태그 안에 link 태그와 함께 해당 링크를 복사한다.

```
p {  
  font-family: 'Roboto', sans-serif;  
}
```

후에 css에서 해당 폰트 이름과 함께 사용한다.

여기까지가 html 의 link 태그를 이용한 웹 폰트 불러오기 방법이었다.

- **Web Fonts Using @font-face**

폰트는 @font-face 를 통해 html에 link 태그 기입없이 css 시트에 바로 적용할 수 있다.

- OTF (OpenType Font)
- TTF (TrueType Font)
- WOFF (Web Open Font Format)
- WOFF2 (Web Open Font Format 2)

폰트는 웹에서 다운이 가능한데 이렇게 4가지 포맷으로 나뉘게 된다. 아래로 갈수록 가장 최신의 포맷이다.
브라우저의 호환을 위해 보통 다 다운 받는다.

```
@font-face {  
  font-family: 'MyParagraphFont';  
  src: url('fonts/Roboto.woff2') format('woff2'),  
       url('fonts/Roboto.woff') format('woff'),  
       url('fonts/Roboto.ttf') format('truetype');  
}
```

폰트를 다운로드한 뒤 위 양식을 css에 적어준다. 모든 포맷을 다 적는다. font family 이름은 자유롭게 지을 수 있다.

```
p {  
  font-family: 'MyParagraphFont', sans-serif;  
}
```

이제 css 에서 사용자정의 이름으로 바로 사용이 가능하다!!

어떤 이점이 있을까? html 의 사이즈를 줄여주고 복잡성을 줄여준다. 또한 css 와 html 을 굳이 오가지 않아도 된다!

대신 다운로드를 해야하는 단점이 존재한다.

오늘의 단어

- succinctly: 간결하게
- dogmatic : 독단적인, 자기주장을 고집하는
- heck : 제기랄, 도대체
- go for sb , sth : ~에 해당되다.
- regression : 회귀; 퇴보