



# Waffle Studio

# Frontend Seminar - 1

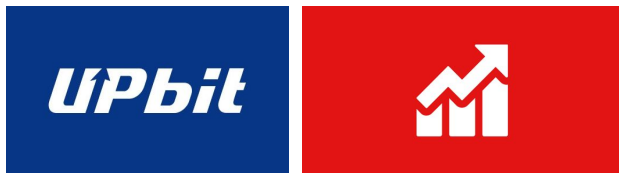
김기완



# Instructor

김기완

- 컴퓨터공학 전공
- 휴학 후 병역 특례 티오 대기 중
- 두나무 프론트엔드 개발자(react, php, jquery...)





# 질문하는 법

세미나 중에는...

- 가급적 채팅으로

그 외에는...

- 선 구글링, 후 질문
- 질문은 웬만하면 깃 이슈에 해주세요.



# Frontend?

- 사용자에게 보이는 영역
- 주로 웹 페이지를 지칭
- **UI/UX**에 대한 이해

신경 쓸 것도 다양합니다

- 크로스 브라우징
- 웹 접근성
- 정보 통신 보조기기

# HTML

- 페이지의 구조, 의미를 부여하는 정적인 언어
- 웹의 구조를 담당
- 이쁘게 만들려고 시도 X
- 온전한 구조를 만드는 데 집중



# CSS

- 크기, 폰트, 레이아웃 등을 지정하여 콘텐츠를 꾸며주는 정적 언어
- 웹의 시각적인 표현을 담당
- 적당한 크기와 아름다운 색상, 원하는 위치를 지정하는데 집중

# Javascript

- 페이지를 동적으로 꾸며주는 역할을 하는 프로그래밍 언어
- 웹의 동적 처리를 담당



# HTML

## 기본 형태

- 태그는 각자의 의미를 가짐
- 열리고 닫히는 태그가 한 쌍
- 닫히는 태그 앞에는 `/` 가 붙음

```
<h1>토끼와 거북이</h1>
```

```
<p>옛날 옛적에...</p>
```

```
<!--아래와 같이 해석됩니다.-->
```

```
<주제1>토끼와 거북이</주제1>
```

```
<문단>옛날 옛적에 ...</문단>
```

# HTML

## 속성(Attribute)와 값(Value)

- 태그의 기능을 확장할 때 사용

```

```

```
<div class="세미나">프론트엔드</div>
```

```
<!--아래와 같이 해석됩니다.-->
```

```
<사진 소스위치="./waffle.png" 대체 텍스트="와플 텍스트" />
```

```
<의미없는분할 태그별명="세미나">프론트엔드</의미없는분할>
```



# HTML

부모와 자식 요쇼

빈 태그

- `/`는 필수가 아님

```
<PARENT>
```

```
<CHILD />
```

```
<CHILD >
```

```
</PARENT>
```





# HTML 문서

## `<html>`

- HTML 문서의 전체 범위
- 웹 브라우저가 해석해야 할 범위 지정

## `<!DOCTYPE>`

- HTML 문서의 해석 방식

```
<!DOCTYPE html>
<html lang="en">
  <head>
    문서의 정보
  </head>
  <body>
    문서의 구조
  </body>
</html>
```



# HTML 문서

## <head>

- HTML 문서의 정보
- 제목, 인코딩 방식 등의 기본 세팅 값

## <body>

- HTML 문서의 구조
- 화면에 나오는 내용의 형태, 레이아웃

  
**<head>**

HTML 문서의 정보

**<title>**

- HTML 문서의 제목
- 브라우저의 탭에서 표시



## <head>

### <meta>

- HTML 문서의 정보를 검색엔진/브라우저에 제공
- 빈 태그

속성	의미	값
<code>charset</code>	문자 인코딩 방식	<code>UTF-8</code> , <code>EUC-KR</code> 등
<code>name</code>	정보의 종류	<code>author</code> , <code>description</code> 등
<code>content</code>	<code>name</code> 의 값을 제공	



## <head>

### <link>

- 외부 문서(특히, **css**)를 연결할 때 사용
- 빈 태그

속성	의미	값
<b>rel</b>	외부 문서와 관계	<b>stylesheet</b> , <b>icon</b> 등
<b>href</b>	외부 문서의 위치	경로



**<head>**

**<style>**

- HTML 문서 내부에서 css 작성

**<script>**

- js 파일을 불러오거나, HTML 문서 내부에서 작성



**<body>**

**<div>**

- 막 쓰는 태그
- 명확한 의미 X
- 단순히 특정 부분을 묶는 역할



## <body>

### <img>

- HTML에 이미지를 삽입할 때 사용
- 빈 태그

속성	의미	값
<b>src</b>	이미지의 경로	URL
<b>alt</b>	이미지의 대체 텍스트	



  
**<body>****<a>**

- 다른 페이지, 같은 페이지, 파일 등 다른 URL로 연결할 수 있는 하이퍼링크 설정

속성	의미	값
<b>href</b>	링크 URL	URL



## **<body>**

이 외에도, **<header>**, **<p>**, **<h1>**,.. 등의 여러가지 다양한 태그가 존재

각 태그를 구조에 맞게 사용하는 것이 중요

[한눈에 보는 HTML 요소\(Elements & Attributes\) 총정리](#)



# CSS

## 선택자

- HTML의 요소를 선택하는 사인(sign)
- 적용할 스타일을 속성, 값으로, 대상을 선택자로 나타냄
- 태그로 찾기 / 클래스로 찾기

```
<div>  
  <h1>제목...</h1>  
  <p class="description">본문...</p>  
</div>
```

```
h1 {  
  color:  red;  
}  
  
.description {  
  color:  blue;  
}
```



# CSS

속성(Properties)과 값(Value)

- 스타일을 지정하는데 사용
- 많이 아는 것이 중요

```
div {  
  color: ■ red; /* 글자색은 빨강 */  
  font-size: 20px; /* 글자 크기는 20px */  
  width: 300px; /* 가로 너비는 300px */  
  margin: 20px; /* 바깥 여백은 20px */  
  padding: 10px 20px; /* 안쪽 여백은 위아래 10px, 좌우 20px */  
  position: absolute; /* 위치는 부모 요소 기준, 흠.. 이게 무슨 뜻일까? */  
}
```



# CSS

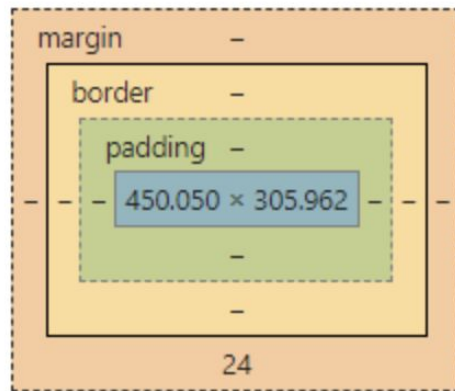
## 적용 방식

- 인라인
- HTML 내부에서 작성
- HTML 외부에서 작성

# CSS

## margin and padding

- 모든 HTML 요소는 하나의 box
- 태그 안의 내용은 파란 부분(contents box)으로





# CSS

## position

- **static**: HTML 문서의 일반적인 **flow**에 따라 배치
- **relative**: **static**에서 지정된 위치에서 얼마나 이동할 지를 지정
- **absolute**: **static**이 아닌 부모 요소를 기준으로 어디에 위치할 지를 지정
- **fixed**: 스크롤과 관계 없이 항상 고정된 위치



## Assignment - 깃헙 따라 만들기





# 참고할만한 사이트

[한눈에 보는 HTML 요소\(Elements & Attributes\) 총정리](#)

[CSS 개요](#)

[CSS 속성](#)