

Open Year Round 운영진 김현지



2주차



프론트 & 백 / 심화

( 목차 )

- **프론트 & 백**

프론트엔드

백엔드

대표적 프레임워크

- **심화**

Flex

Grid

01  
02  
03  
04  
05  
06  
07  
08  
09  
10

( 프론트<sup>1</sup> & 백 )

WEB

프론트엔드 + 백엔드



+



# ( 1-1 프론트엔드 )

사용자가 볼 수 있는 화면

사용자 인터페이스(UI)

웹을 쉽게 사용할 수 있도록 기술적으로 구현

사용자 경험(UX) 최적화에 초점

HTML

CSS

JavaScript

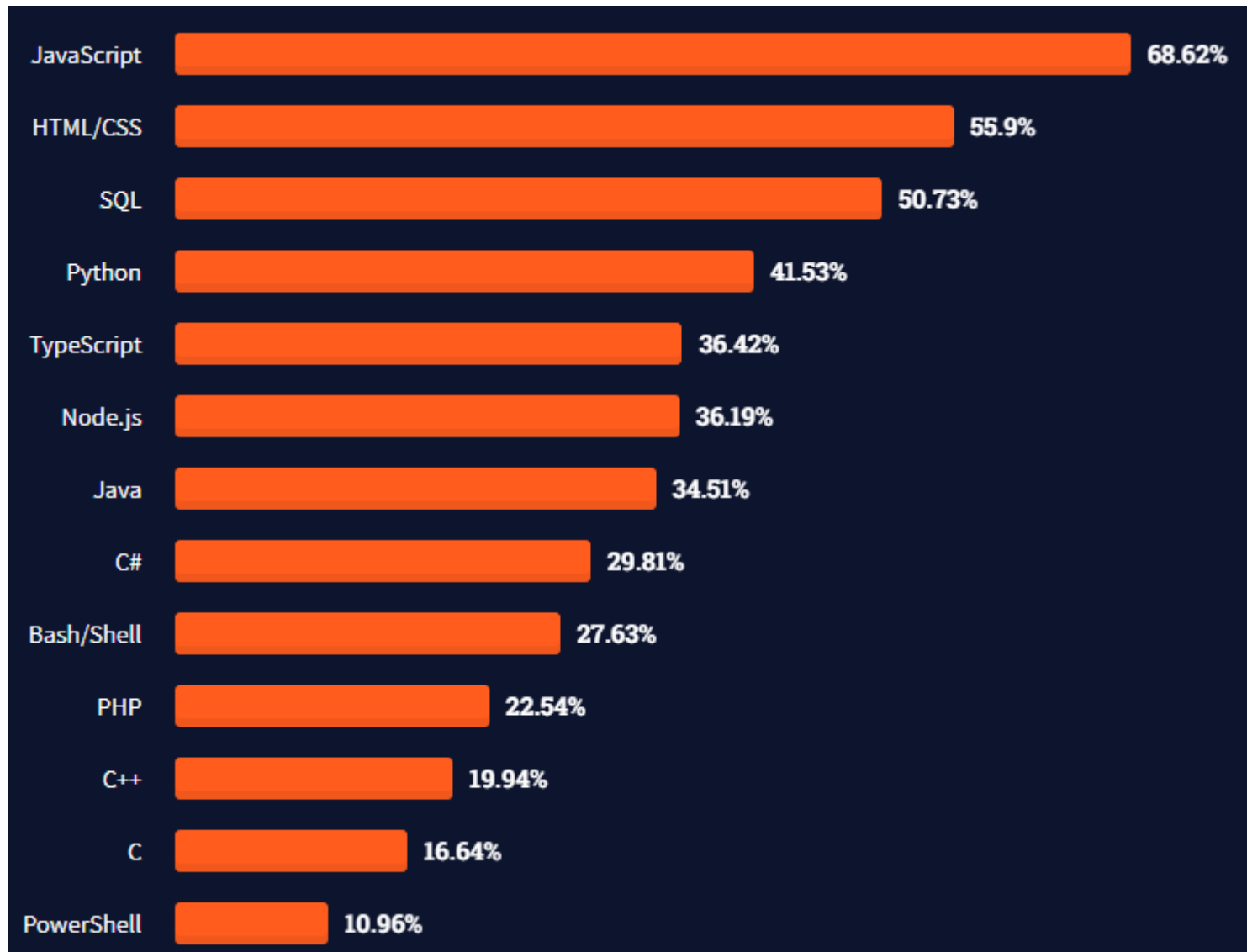
( 1-2 백엔드 )

사용자가 볼 수 없는 서버, DB를 관리

실질적으로 사용자들이 요청하는 정보 제공

Java  
PHP  
Ruby  
Python

# ( 언어의 선호도 )



( 1-3 대표적 프레임워크 )

웹 개발을 쉽고 간편하게 할 수 있도록 도와주는 도구

프론트엔드 + 백엔드



+



## Angular

- 타입스크립트 기반 오픈소스 프론트엔드 웹 애플리케이션 프레임워크
- 프로젝트 생성, 테스트, 빌드, 배포 모두 가능
- 틀이 정해져 있어 그에 맞춰 코드 작성 → 쉽게 이해, 수정 가능

## React

- 선언형 자바스크립트 라이브러리, 컴포넌트 기반의 오픈소스 라이브러리
- 스택을 마음대로 골라 사용 가능 → 유연

## Vue.js

- UI를 만들기 위한 오픈소스 자바스크립트 프레임워크
- 자바스크립트 라이브러리를 사용하는 프로젝트에 뷰를 도입하기 쉬움



## Express

- Node.js용 무료 오픈소스 프레임워크 → API 제작, 웹 애플리케이션 구축
- 노드의 서버 개발을 쉽고 강력하게 만들 수 있도록 도와줌
- 노드의 기능을 그대로 활용할 수 있음

## Spring

- 자바의 오픈소스 애플리케이션 프레임워크
- 규모가 큼 → 성능, 안정성 탁월 → 많은 트래픽 수용 가능
- 확장성 좋음 → 지원하지 않은 라이브러리도 사용가능

## Django

- 파이썬의 웹 프레임워크
- 로그인, 인증, 파싱 등 기본적인 기능 탑재
- 커스텀은 어렵지만 개발 생산성 높음

## Laravel

- PHP의 웹 프레임워크
- 캐싱, 라우팅, 인증, 세션 웹 개발에서 사용되는 작업 간단
- PHP의 선호도가 감소하면서 사용 감소

## Flask

## Rails

# Flex

한 방향 레이아웃 시스템(1차원)

Flexible Box, Flexbox

레이아웃 배치 전용 기능

```
.container {  
  display: flex;  
}
```

# 기본 HTML 구조

```
<div class="container">  
  <div class="item">helloflex</div>  
  <div class="item">abc</div>  
  <div class="item">helloflex</div>  
</div>
```

부모 요소 container → Flex Container

자식 요소 item → Flex Item

컨테이너가 Flex의 영향을 받는 전체 공간, 설정된 속성에 따라 각각의 아이템들이 어떤 형태로 배치

## 속성

### 컨테이너에 적용하는 속성

- flex-direction: 배치 방향 설정
- flex-wrap: 줄넘김 처리 설정
- flex-flow: flex-direction과 flex-wrap을 한 번에 지정
- justify-content: 메인축 방향 정렬
- align-items: 수직축 방향 정렬
- align-content: 여러 행 정렬

# 속성

## 아이템에 적용하는 속성

- flex-basis: 유연한 박스의 기본 영역
- flex-grow: 유연하게 늘리기
- flex-shrink: 유연하게 줄이기
- flex: basis, grow, shrink를 한 번에 사용하는 축약형 속성
- align-self: 수직축으로 아이템 정렬
- order: 배치 순서
- z-index: z축 정렬

# Grid

```
.container {  
  display: grid;  
}
```

두 방향(가로-세로) 레이아웃 시스템(2차원)

HTML 기본구조 → 컨테이너, 아이템

부모요소 container → Grid Container

자식 요소 item → Grid Item

컨테이너가 Grid의 영향을 받는 전체 공간, 설정된 속성에 따라 각각의 아이템들이 어떤 형태로 배치

## 속성

- grid-template-rows/columns: 그리드 형태 정의
- row-gap/column-gap/gap: 간격 만들기
- grid-auto-rows/columns: 그리드 형태를 자동으로 정의
- grid-column/row-start/end/-: 각 셀의 영역 지정
- grid-template-areas: 영역 이름으로 그리드 정의
- grid-auto-flow: 자동 배치



## 속성

- align/justify-items: 세로/가로 방향 정렬
- place-items: align-items, justify-items 한 번에 사용
- align/justify-content: 아이템 그룹 세로/가로 정렬
- place-content: 위 2개 한 번에 사용
- align/justify-self: 개별 아이템 세로/가로 정렬
- place-self: 위 2개 한 번에 사용
- order: 배치 순서
- z-index: z축 정렬

( end )

thank you.