강의 소개 및 강의 준비

3 사전 과제의 존재 이유와 팁

사전 과제의 존재 이유

문제 해결 능력

- 얼마나 문제를 **정확하고 효율적**으로 풀었는가?
- 제안한 방법이 얼마나 **확장성** 있고, **유지보수가 용이**한가?

협업 능력

- **변수 네이밍**이 일관성 있고, 정보를 잘 전달하는가?
- 적절한 도구를 활용해 (prettier, eslint) **코딩 컨벤션**을 준수했는가?
- 모듈화가 잘 되어 있는, **파악하기 쉬운** 코드인가?

기술적 역량

- 적절한 **문법**을 사용해 구현했는가?
- **Edge Case, 에러 처리**에 대한 고려가 되어 있는가?
- **성능 최적화**가 되어 있는가?

사전 과제 풀이 팁

1. 섣부른 구현은 금물

의사코드 (pseudocode) 혹은 주석, 손그림 등으로 **코드 흐름을 미리 생각**해보고 구현에 들어가기

```
// 1. 진입 시 input focus 구현 (1. <input autofocus /> 2. window.addEventListener)
const $idInput = document.getElementById('idInput')
window.addEventListener('load', $idInput.focus())

// 2. 이메일 validation 로직 구현
// 유효하지 않을 경우, 이메일 input의 style이 달라져야 하고, 에러 메세지가 나타나야 함
// 2-1. 비어있을 경우: "필수 정보입니다"
// 2-2. 패턴에 맞지 않을 경우: "5~20자의 영문 소문자, 숫자와 특수기호(_),(-)만 사용 가능합니다."
```

사전 과제 풀이 팁

2. 아는 것을 잘 풀기

최대한 많은 문제를 정확하지 않게 구현하는 것 < 아는 문제들을 정확하고 완벽하게 구현하는 것

사전 과제 풀이 팁

3. 코드 가독성과 재사용성 고려하기

잘 돌아가기만 하면 되는 코드 (X) 코드 리뷰를 진행하는 코드 (O)

적절한 네이밍과 모듈화 + **반복되는 로직은 재사용** 가능하도록 확장해 활용

```
import Footer from './components/Footer'
import Form from './components/Form'
import FontControlBox from './components/FormControlBox'
import Modal from './components/Modal'
```

사전 과제 풀이 팁

4. HTML5, CSS3, ES6 문법 숙지하기

문법을 숙지하지 않으면 적합한 선택지들을 충분히 생각해내기 어려움

사전 과제 풀이 팁

5. 개발자 도구를 활용한 디버깅

콘솔 창과 debugger을 활용한 디버깅 역량이
구현 과정에서 막히는 부분을 잘 해결할 수 있는지를 결정 지음