

HTML부터 React까지

간지나는 멘토이고싶은 감자 POTATO 김난슬

멘토링시작 전 코멘트

다들 과제 어땠나요?

간단한 오류, 조작법, 내 상황이 막막함, 챗지피티가 못고침, 내 약점이 뭔지 모르겠음

-> 멘토에게 물어보기

깃헙데스크탑 오류, css충돌, 개발툴잡기, 개념질문(이미지 어떻게 넣음?)

-> 챗지피티, 또는 간단한 영상이나 블로그로 추가개념학습 (진짜 이게 나한테 물어보는것보다 훨훨 빠름)

(이런 오류를 고쳤는데 그럼 내 약점은 뭐고 나 뭘공부해야됨?은 멘토에게)

그러나????????님들 이제 제대로 공부한지 일주일됨ㅋㅋㅋ아직 질문구분못하는게 당연.

(그러나,,,챗지피티에게 물어보는건 적극활용하자...나보다 더 잘가르침 ㄹㅇㄹㅇ)

그리고 여러분들 질문하는 걸 보고 제가 여러분들 약점과 상황을 파악하기도하니까 최대한 멘토링때는 많이많이 해서 뽕뽑자.

샵질해도 됩니다. 괜찮아요.

아는 것이 많으면 샅질양이 줄어들긴 하지만! (이 사실을 뼈저리게 이번과제로 느꼈을거라 생각함)

그 샅질로 교훈을 얻었다면 내가 아는 것이 하나 추가되었구나 좋다ㅎㅎ알면 된거지 (가성비생각금지)

하고 다시는 그 샅질을 안하려고 노력하면 됩니다.

그렇게 성장하는거라고 저는 생각해요.

복습테스트 보자~!~!

이거 너네 멘토링 끝나고 플젝할때 다른 팀원들 재촉할때
이에진시점으로 찍은거임ㅋㅋㅋㅋ
님들 앞으로 열심히 안해오면 내가 저 표정으로 학교에서
계속 쫓아다님ㅋㅋㅋ억ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ



과제 제대로 해보니까 다른 멘토링과 다르게 정말 빠세고 알차죠?^^^ 그래서 오늘은 제가 더 준비했어요^^^

**괜찮아. 너넨 할 수 있어. 왜냐 우리 4주 쉬고 10/31에 만나욐 >< 기한 넉넉하죵? 허허
10/31부터 바로 어려운 JS비동기프로그래밍 & 리액트할꼬양ㅎㅎㅎ이거 열심히 해오세욐!**

-
- 01 CSS 주의사항 심화편
 - 02 CSS 반응형디자인 with media query
리액트 살짝 짹먹?
 - 03 JS 기본
 - 04 JS 이벤트처리
 - 05 JS DOM 조작
 - 06 과제 안내

CSS 주의사항

일부러 부딪히면서 배우라고 안알려주긴했는데...그래도 몇개는 알려주는게 더 편할수도 있다는 생각에 배워봅시다. css꿀팁!이라고생각하면 될듯
다들 하면서 많이 힘들었죠??^^^

1. 우선순위
2. 상속
3. 마진겹침
4. 디버깅방법

사실 이렇게 간단한 퍼블리싱말고 진짜 실무에서는 스타일을 명확하게 하기위해
css module, style-component, bem방법론, scss(서비스자체가 복잡해서 css도 복잡할
때)을 이용하긴함ㅋㅋ근데 이거해도 충돌은 일어나긴해~

CSS 주의사항 - 우선순위 Cascading

1. 인라인 > id > class > 태그 선택자 순
2. !important 를 쓰면 가장 우선순위 높아짐 (but 유지보수가 어려워짐 -> 남용금지, 최후의 보류)
3. 구체적인 선택자 사용 (구체적일수록 더 우선순위 높아짐)
4. 중복된 스타일 제거 (**개발자도구**, 피닉스코드, purgecss)

피닉스코드는 그 태그에 어떤 스타일이 적용되었는지 확인할때, purgecss는 중복된 스타일을 알아서 제거해주는건데 정확성이 그닥..그냥 이거 다 개발자도구로 할 수 있으니까 개발자도구 적극활용

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h1 {
        color: black;
      }
      #header-title {
        color: purple;
      }
      .highlight {
        color: orange;
      }
    </style>
  </head>
  <body>
    <h1 id="header-title" class="highlight">Hello, World!</h1>
  </body>
</html>
```

즐거운 퀴즈시간 ^~^

다음 h1 요소의 텍스트 색상은 무엇일까요~?

CSS 주의사항 - 상속

대부분의 충돌이 여기에서 일어남! 상속은 display, position, flex만큼이나 진짜 개개개개개 중요!!!!!!!
리액트에서는 컴포넌트끼리 깊게 중첩되는 경우가 많기 때문에 상속이 특히특히특히 중!!!!!!요!!!!!!!!!!!!!!

텍스트형식과 관련된 속성들만 상속됨	vs	레이아웃&박스모델 속성들은 상속안됨
ex) color, font-family, font-size, line-height, text-align		ex) margin, padding, border, background, width&height

그럼 결론이 뭐임
-> 강 충돌 직접 많이 겪어보면서 부모태그에는 최대한 공통적인것만 하고
자식태그에 노가다로 하나하나 설정하는수밖에...
프론트가 노가다로 불리는 이유 (근데 충돌겪는것보단 나은듯ㅋ)

CSS 주의사항 - 마진겍침

이건가? 한번 의심해보고 적용해보는 정도로만 이해하고 나중에 떠올릴 수 있기만 하면 됨 (암기금지)

마진겍침은 아주 좋은 참고영상이 있다! 예시코드를 보면서 직접 이해해보자

은근 몰라서 오류나는 것

1. position: absolute / fixed가 지정된 요소는

자동으로 display: block + 반응형 디자인에 해로우니 신중히 사용

추가적으로 기본적인 인라인 블록요소가 뭔지 알아두는게 굉장히 도움이 됨 ex) svg태그는 인라인이 디폴트값

2. 반응형디자인에 유리한 디자인, 즉 유지보수가 가능한 css style은 따로 있다.

ex) flexbox 적극이용,

position relative(부모)를 이용해 기본위치 설정 후 자식요소의 absolute위치 잡을때 유용 (카드ui 깃헙 resources에 올려놨으니확인)

-> 특정 요소가 부모 요소 내에서는 정밀하게 배치되어야할때 (근데 정렬 잘 안바뀌는)

-> 그러나 애도 가/세로정렬 변하는 반응형에서는 최악임 그냥 웬만하면 flex쓰세요쩍발.....근데 또 정렬이 그렇게 안변할때는 이거쓰자
괜히 flex가 다루기 어려워서 게임(froggy~)으로 나오는게 아님 오버엔지니어링은 또다른 죄.

CSS 주의사항 - 디버깅방법

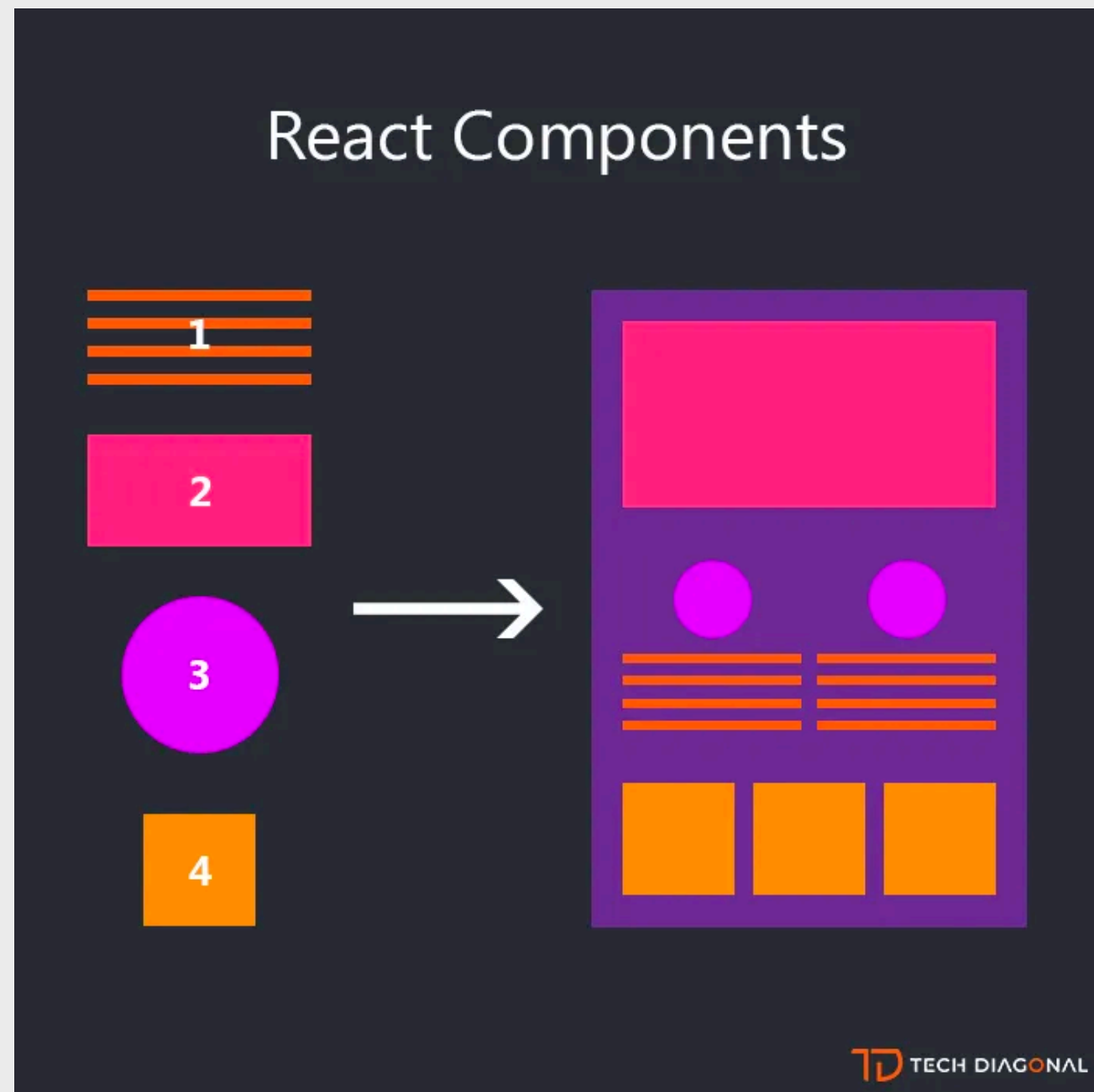
1. 개발자 도구 적극활용 (요소검사, 한줄검사, 상속 체크, 충돌지점찾기)
2. border속성이용 (마진겹침, 이 요소가 어디까지 영향을 미치나 체크하기 좋음)

```
.debug {  
    border: 1px solid red; /* 요소의 경계가 명확해짐 */  
}
```

class = "debug"만 계속 추가하면 그 요소 파악가능!

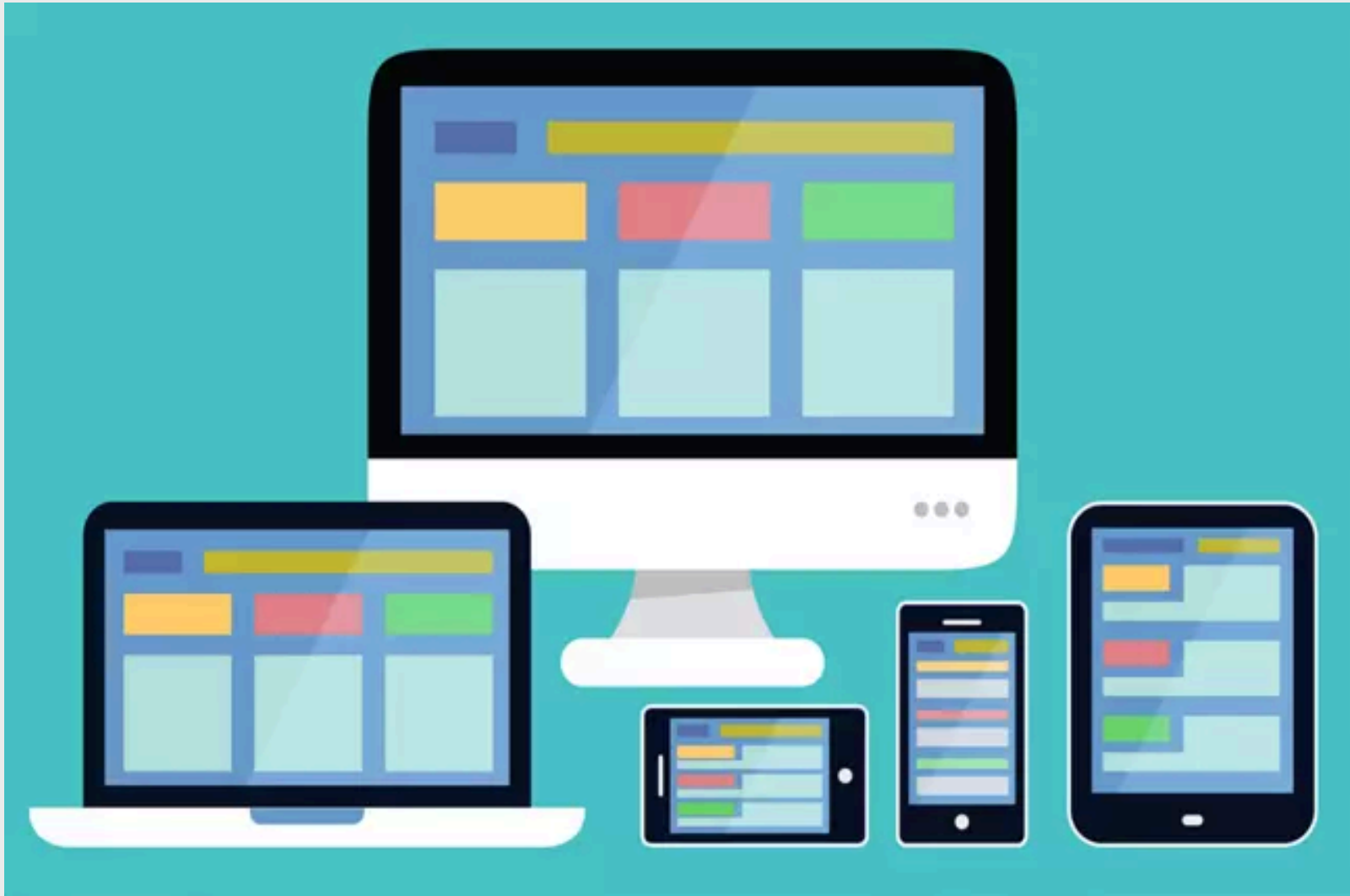
리액트를 써야하는 이유 - CSS가 덜 깨같아짐 사랑스러워짐

컴포넌트 기반 아키텍처



CSS, 요소 반복작업 안해도 됨!!!우와!!!!

반응형디자인이 왜 중요할까?



UX(사용자경험) & 접근성 향상

근데 진짜 쉽지않음....ㅋㅋㅋㅋ

why? 내가 데탑기준으로 잡아놓은 css단위, position, display 다 조정해야됨
어떤 css단위와 flex를 어떻게 잡아야할지를 한번 생각해보자!

CSS 반응형 디자인

직접 영상을 보고 따라해보자!

JS 기본

음..그전에 JS는 어떻게 공부하면 되냐면요...그냥 이렇게 있구나 하고 이렇게 해야 좋은 코드구나하고 따라하면 됩니다ㅋㅋㅋ 작동방식을 보고 함수가 이렇게 작동하는구나를 이해만 하면 됨! 문법 집착금지 어차피 챗지피티가 다 짜줌 특히 JS는. css랑 약간 커스텀하는게 달라요 JS의 어려운 부분은 리액트에서 진짜로 시작됩니다.... 지금은 쉬움!

변수나 함수, 객체, 클래스에 대해서는 대부분 이미 어느정도는 알고있.....지.....?

-> 그냥 추가적으로 혼자 공부하고 자바스크립트는 이렇게 작동하는구나하는 **차이점위주로** 잡아가면 됨

과제안내

난 커스텀하려면 처음부터 하나하나 다 꼼꼼히 공부하고 이해할거야' 마인드의 열정은 좋지만 이 개발파트에서는 그렇게 성장에 도움이 되지않아요... 이걸 내신공부가 아닙니다. 철저히 개발은 “직접” 해보면서 부딪히면서 성장합니다. 커스텀을 해보다가 이해가 안되는 문구나 개념이 생기면 그때그때 다시 회귀해서 그 부분에 대한 이해를 하고 다시 개발을 하는 방향이 맞습니다. 아예 감도 못잡을 정도로 기본적인 지식이 부족한 경우는 좀 오래 걸리겠지만(애도 기능은 다 어떻게든 구현가능하긴함) 우리는 기본적인 지식이 있잖아요?

우선순위

1. 담당 금쪽이 코드리뷰
2. 마진검침3 영상보기
3. 미디어쿼리실습&PPT보면서 자기약점 반드시 채울것
4. 코드 리뷰반영해서 유지보수를 생각한 코드로 수정
5. **layout페이지만** 미디어쿼리 적용해서 반응형으로 수정
6. 자바스크립트 강의 들어오기 (복습테스트때 체크 예정)

각자 담당 금쪽이잊지않았죠?^^ 협업에 쓰이는 소프트스킬은 기본이자 핵심!

- 피그마 devmode 적극이용
 - 챗지피티로 틀잡고 커스텀해보는거 연습해볼것
 - 이걸 어떻게 해야할지 감조차 안올시 바로 질문하는것이 노베상황에서 더 효율적
 - 커밋, 이슈 컨벤션 적극이용
 - 복습할때는 인덱스를 기억해뒀다가 그 인덱스를 그때그때 찾는 느낌으로 공부하자!
-> 어차피 자주 찾다보면 자연스럽게 외워짐, 외우려고 준비한 노력 더 소중히 쓰자
- 절!!!대!!! 페이지 하나당 커밋 한개라는 생각을 제발..버리세요...물론 간단하긴하더라도 커밋은 잘게 쪼갤수록 이득이라고 저는 생각해요...**