 # 리액트란?

- React 는 UI 를 구현하는 JavaScript 라이브러리
- 페이스북에서 제공해주는 프론트엔드 라이브러리
- 웹 앱(Web App) 또는 네이티브 앱(Native App)
- 유지보수를 쉽게 , DOM 관리
- 성능최적화 쉽게
- 컴포넌트에 집중
- 대부분 공식 라이브러리가 없음 (높은 자유도)
- 사용하는 큰 기업이 많음
- 자바스크립트 친화적 es6 기반로 배우기가 쉽다

 # 앵귤러

- 자체 내장된 기능이 많음
- 다양한 공식 라이브러리가 존재
- TypeScript 가 거의 강제적
- 성숙하나 인지도 측면에서는 성장 단계
- 배우기가 어렵다
- Directive 사용

```
<ul class="heroes">
  <li *ngFor="let item in list">
    [class.selected]="item === selectedItem" (click)="onSelect(item)"
  </li>
</ul>
```

 # 뷰

- 사용하기 쉬움 (리액트와 앵귤러보다)
- 웹팩없이 사용가능
- HTML 을 템플릿처럼 활용
- 앵귤러와 비슷한 구조 (ngFor -> v-for)
- Directive 사용

```
<ol>
  <li v-for="item in list">{{ item }}</li>
</ol>
```

react 에 필요한 es6(ecmascript) 정리하기

1. let / const

let 과 const 의 차이점은 변수의 immutable 여부
let 은 변수에 재할당이 가능하지만,
const 는 변수 재선언, 재할당 모두 불가능
{} scope

2. 템플릿 리터럴(Template literal) 새로운 문자열 표기법

1) backtick (`)
2) 변수 처리는 ${변수}

```
const a = 10
const b = 20
const str = `${a} + ${b} `
console.log(str)
```

3. 삼항연산자와 &&연산자 , ||연산자

조건 ? true 일때 : false 일때
같다 === 사용
true && 결과
(false , null , undefined) || 결과

4. 화살표 함수

```
function(){}
 () => { ... } # 매개변수가 없을 경우
function(x){}
 x => { ... } # 매개변수가 한 개인 경우, 소괄호를 생략
function(x,y){}
(x, y) => { ... } # 매개변수가 여러 개인 경우, 소괄호를 생략할 수 없다.
```

#. 배열.메서드() 중 불변성 유지

5. push

배열 뒷부분에 값을 삽입 (배열의 값이 변경된다)

```
let arr = [ 1, 2, 3, 4 ];
arr.push( 5 );
console.log( arr ); // [ 1, 2, 3, 4, 5 ]
```

6. concat

다수의 배열을 합치고 병합된 배열의 사본을 반환 **기존의 배열은 건드리지 않음 **

```
const arr = [0, 1, 2];
const arr1 = arr.concat(3); // 결과 [0, 1, 2, 3]
const arr2 = arr.concat();
```

7. slice

```
slice( startIndex, endIndex)
배열의 startIndex 부터 endIndex 까지(endIndex 는 불포함)에 대한 shallow copy 를 새로운 배열 객체로 반환
**기존의 배열은 건드리지 않음 **
const arr = [ 1, 2, 3, 4, 5, 6, 7 ];
const newArr = arr.slice( 3, 6 );
```

8. map

```
배열.map((요소, 인덱스 ) => {
    return 요소
});
```

8. filter

```
filter 도 map 함수와 마찬가지로 콜백함수를 인자로 받는데 모든 원소를 한번씩 돌리면서 콜백함수의 몸체부분에서
true 를 반환하는 원소들만 걸러줌  - 결과 배열
const arr = [0, 1, 2, 3, 4, 5];
const newArr = arr.filter( () =>  조건  )
```

9. find / findIndex

```
const newArr = arr.find( () => 조건 )  결과 배열의 값
```

10. 객체

```
객체를 선언 할 때에는 이렇게 { } 문자 안에 원하는 값들을 넣준다
키: 원하는 값
const dog = { name: '멍멍이', age: 2  };
console.log(dog.name);
console.log(dog.age);
```

11. 비구조할당

```
const object = { a: 1, b: 2 , c:10 };
const { a, b , c} = object;
console.log(a); // 1
console.log(b); // 2
console.log(c); // 10 react class 컴포넌트 예)
```

12. spread... (전개 연산자)

```
const ani = ['개', '고양이', '참새'];
const ani1 = [...ani, '비둘기'];
```

13. reduce

```
배열.reduce((누적값, 현잿값, 인덱스, 요소) => {
return 결과
}, 초기값)  초기값 생략하면 1
```

14. forEach

```
배열.forEach( (요소) => { } )
```

15. 검색에 필요한 명령어

```
string.indexOf(찾을문자열)
정규표현식을 사용한 match() 함수
```