프론트엔드 면접문제\_강현주\_20200703

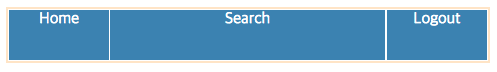
1. FLEX 레이아웃과 GRID 레이아웃을 설명하고 예제를 만드시오.

* **FLEX 레이아웃** : 뷰포트나 요소의 크기가 불명확하거나 동적으로 변할 때에도 효율적으로 요소를 배치, 정렬, 분산할 수 있는 방법을 제공하는 CSS3의 새로운 레이아웃 방식이다. 이 Flexbox의 장점을 한 마디로 표현하면 ‘복잡한 계산 없이 요소의 크기와 순서를 유연하게 배치할 수 있다'라고 할 수 있다. 정렬, 방향, 순서, 크기 등을 유연하게 조절할 수 있기 때문에 별도의 분기 처리를 줄일 수 있고, CSS만으로 다양한 레이아웃을 구현할 수 있다. CSS규칙에 따라 부모 요소 (Flex Container)와 자식 요소(Flex Item)로 작업 한다.
  + **FLEX 레이아웃 예제 )**

코드

****

크롬 화면

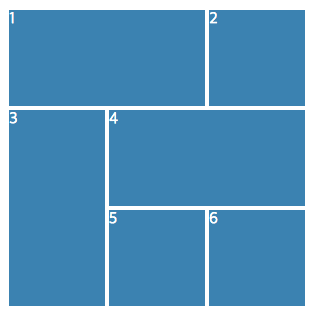


* **GRID 레이아웃** : Grid 레이아웃은 웹사이트 디자인의 기본 요소이자 1차원 시스템인 Flexbox와는 다르게 열과 행 모두 다룰 수 있는 2차원 시스템이며 레이아웃을 보다 수월하게 구성할 수 있도록 도와준다. CSS규칙에 따라 부모 요소 (Grid Container)와 자식 요소(Grid Item)로 작업 한다. 아직 모든 브라우저에서 사용할 수 있진 않다.
  + **GRID 레이아웃 예제 )**

코드



크롬 화면



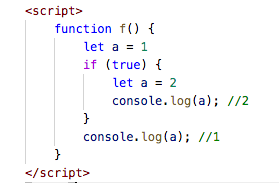
출처 :

1. <https://webzz.tistory.com/655>
2. <https://www.vobour.com/-%EB%94%94%EC%9E%90%EC%9D%B8-5%EB%B6%84-%EC%95%88%EC%97%90-flexbox-%EB%B0%B0%EC%9A%B0%EA%B8%B0>

2. ES6문법을 10가지이상 설명하고 각 예제를 만드시오.

1. **let / const** : 블록 유효 범위를 갖는 새로운 변수 선언 방법을 지원한다. let 은 var와 유사. const는 재할당 및 재선언이 불가능. var의 유효 범위는 전체 외부 함수까지지만 let은 변수를 선언한 블록과 그 내부 블록들에서 유효하다. const은 상수 let은 변수

* 예제)



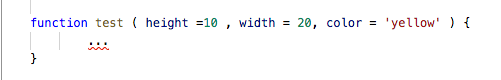
1. **템플릿 / 백틱** : 백틱( `)을 이용하여 문자열을 + 기호 없이 간단히 처리할 수 있다. 또한 백틱( ` ) 안에서는 멀티 라인도 자유롭게 사용할 수 있다.

* 예제)



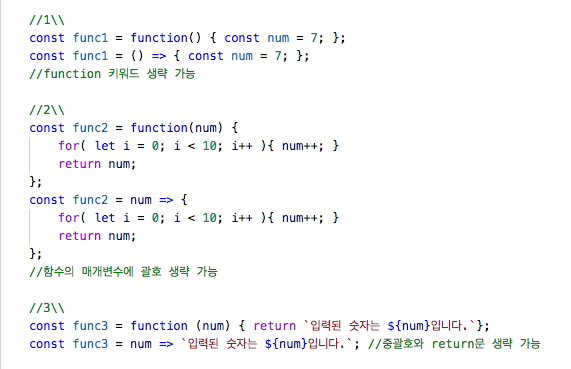
1. **함수의 기본 매개 변수값 지정** : ES6 의 함수는 매개변수에 기본값(default)을 지정해 줄 수 있다.

* 예제)



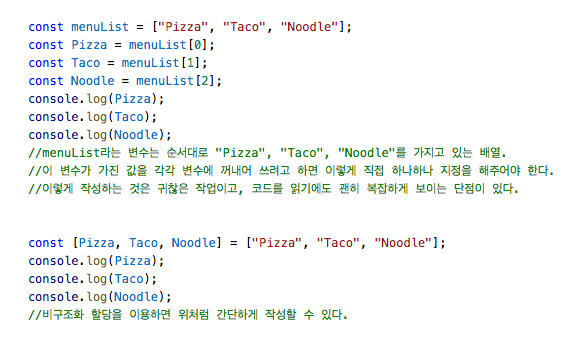
1. **화살표 함수** : 함수를 표현하는 것이 더 간단해졌고, this가 정적으로 묶인다. 객체 생성자로 사용할 수 없고 arguments 변수를 사용할 수 없다.

* 예제)



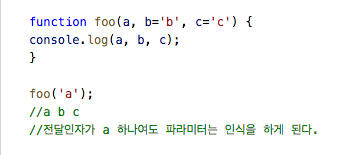
1. **비구조화 할당** : 배열이나 객체의 요소를 해체하여 별개의 변수로 추출할 수 있도록 하는 것이다. 할당 실패에 유연하며, 실패 시 undefined값이 자동할당 된다. 또한 객체의 속성 값도 자동으로 검색하여 바인딩해준다.

* 예제)



1. **Default Parameter** : 함수 파라미터의 기본값을 설정할 수 있다.

* 예제)



1. **모듈** : 모듈이란 애플리케이션을 구성하는 개별적 요소로서 재사용 가능한 코드 조각을 말한다. 모듈은 세부 사항을 캡슐화하고 공개가 필요한 API만을 외부에 노출한다.
2. **클래스** : 클래스는 포로토타입 기반 객체지향 패턴을 더 쉽게 사용할 수 있는 대체재이다. 클래스 패턴 생성을 더 쉽고 단순하게 생성할 수 있어서 사용하기도 편하고 상호운용성도 증가된다.
3. **spread operator** : spread operator를 사용하면 배열이나 문자열과 같이 반복 가능한 문자를 0개 이상의 인수 (함수로 호출할 경우) 또는 요소 (배열 리터럴의 경우)로 확장하여, 0개 이상의 키-값의 쌍으로 객체로 확장시킬 수 있다.
4. **객체 리터럴** : 객체 리터럴은 선언문에서 프로토타입 설정, 선언을 위한 단축 표기법, 메서드 정의, super 클래스 호출 및 동적 속성명을 지원하도록 향상 되었다. 그에 따라 객체 리터럴 및 클래스 선언이 더 밀접되어져, 객체기반 설계가 더 편리해졌다.

출처 :

1. <https://velog.io/@decody/ES6-Sheetsheet>
2. <https://jsdev.kr/t/es6/2944>
3. <https://noahlogs.tistory.com/26>
4. <https://happyer16.tistory.com/entry/ES6-%EB%AC%B8%EB%B2%95-%EA%B0%9C%EC%9A%94>

3. PROPS와 STATE는 언제 쓰는가.

* **PROPS** : PROPS는 컴포넌트에서 사용할 데이터 중 변경되지 않는 데이터를 다룰때 사용한다. 부모 컴포넌트에서 자식 컴포넌트로 데이터를 전달할 때 사용된다. PROPS는 변경이 불가능하다.
* **STATE** : 컴포넌트에서 관리하는 상태 값으로 유동적인 데이터를 다룰 때, state를 사용한다. state는 변경이 가능하고 변경할 때는 setState메서드를 사용해 상태를 변경한다. setState는 비동기로 동작하며 동작완료에 대한 콜백을 설정할 수 있다.

출처 :

1. <https://velog.io/@kyusung/%EB%A6%AC%EC%95%A1%ED%8A%B8-%EA%B5%90%EA%B3%BC%EC%84%9C-React-%EC%BB%B4%ED%8F%AC%EB%84%8C%ED%8A%B8%EC%99%80-%EC%83%81%ED%83%9C-%EA%B0%9D%EC%B2%B4>