**1. FLEX레이아웃과 GRID 레이아웃을 설명하고 예제를 만드시오.**

**1) FELX 레이아웃**

FLEX는 엘리먼트들의 크기나 위치를 쉽게 잡아주는 도구이다. 엘리먼트들의 크기가 불분명하거나 동적인 경우에도 각 엘리먼트를 정렬할 수 있는 효율적인 방법을 제공한다.

holyGrailLayout.html

|  |
| --- |
| <!doctype>  <html>  <head>  <meta charset="utf-8">  <style>  .container{  display: flex;  flex-direction: column;  }  header{  border-bottom:1px solid gray;  padding-left:20px;  }  footer{  border-top:1px solid gray;  padding:20px;  text-align: center;  }  .content{  display:flex;  }  .content nav{  border-right:1px solid gray;  }  .content aside{  border-left:1px solid gray;  }  nav, aside{  flex-basis: 150px;  flex-shrink: 0;  }  main{  padding:10px;  }  </style>  </head>  <body>  <div class="container">  <header>  <h1>생활코딩</h1>  </header>  <section class="content">  <nav>  <ul>  <li>html</li>  <li>css</li>  <li>javascript</li>  </ul>  </nav>  <main>  Lorem ipsum dolor sit amet, consectetur adipisicing elit. Reiciendis veniam totam labore ipsum,  </main>  <aside>  AD  </aside>  </section>  <footer>  <a href="https://opentutorials.org/course/1">홈페이지</a>  </footer>  </div>  </body>  </html> |

출처: https://opentutorials.org/course/2418/13526

**2) GRID 레이아웃**

FLEX는 한방향 레이아웃 시스템이지만 GRID는 두방향(가로-세로) 2차원 레이아웃 시스템이다. 따라서 FLEX보다 좀 더 복합적인 레이아웃 표현이 가능하다.

출처: <https://studiomeal.com/archives/533>

**2. ES6문법을 10가지이상 설명하고 각 예제를 만드시오.**

**1) 화살표 함수** : 익명함수의 단축 표현, 생성자로 사용 불가, 제너레이터로 사용 불가

var square = function(x) { return x\*x; };

var square = (x) => { return x\* x; };

**2) 나머지 매개변수** : 함수의 인자가 들어가는 부분에 …을 입력하면 그만큼의 인수를 배열로 받을 수 있다. 화살표 함수에서도 사용할 수 있다.

|  |  |
| --- | --- |
| function f(a, b, …args){  console.log(a, b, args)  }  f(1, 2, 3, 4, 5, 6)  결과값  1, 2, [3, 4, 5, 6] | var sum = (…args) => {  for (var i=0, s=0; i<args.length; i++){  s\_= args[i];  return s;  }  sum(1, 2, 3, 4, 5);  결과값  15 |

**3) 이터레이터와 for-of** : 반복처리를 단계별로 제어할 수 있다. for-of로 전용반복문을 쓴다. iterable 객체를 생성하고, 이터레이터를 받아와서, next()로 순회한다.

출처: https://jeong-pro.tistory.com/121 [기본기를 쌓는 정아마추어 코딩블로그]

**3. props 와 state는 언제 쓰는가.**

컴포넌트의 속성을 설정할 때는 props를 사용한다.

우리가 바꿀 데이터는 state에 저장한다.