

UI 구현 이론

1. 아래의 그림을 보고 공간 분할 태그의 특성 중 블록(block) 과 인라인(inline)의 대해서 설명하시오.



① 블록(block) 공간 분할

블록 요소는 기본적으로 가로폭 전체를 차지하며, 수직으로 쌓입니다.

블록 요소는 자동으로 줄바꿈이 됩니다.

블록 요소의 크기(width, height)는 지정하지 않으면 부모 요소의 크기를 따라갑니다.

div, h1, p, ul, ol, li 등이 대표적인 블록 요소입니다.

② 인라인(inline) 공간 분할

인라인 요소는 콘텐츠의 일부분만 차지하며, 가로폭을 따로 지정하지 않으면 내용물만큼의 크기를 가집니다.

인라인 요소는 수평으로 나열됩니다.

인라인 요소는 자동으로 줄바꿈되지 않으며, 요소의 크기가 변해도 줄바꿈이 되지 않습니다.

a, span, img, input, button 등이 대표적인 인라인 요소입니다.

2. 아래 그림의 예는 iframe을 사용하여 화면을 구성한 예이다. 왜 iframe을 사용하는지 설명하고, iframe을 사용하는 코드와 방법을 설명하여라.



① iframe을 사용하는 이유:

iframe(inline frame)은 HTML 문서 안에 다른 HTML 문서를 삽입할 수 있는 태그입니다. 즉, 웹페이지 안에서 또 다른 웹페이지를 표시하기 위해서 사용합니다. iframe을 사용하면 웹페이지에서 외부 서버의 콘텐츠를 삽입할 수 있습니다. 예를 들어, 다른 사이트의 동영상이나 지도를 삽입할 수 있습니다. 또한, iframe을 이용하여 광고나 실시간 정보를 표시하는 등의 용도로도 사용됩니다. 하지만, iframe은 보안상의 문제가 있을 수 있

습니다. iframe을 이용하여 다른 사이트의 콘텐츠를 삽입하면, 해당 사이트에서는 자신의 콘텐츠가 무단으로 사용된 것으로 인식할 수 있습니다. 따라서, iframe을 사용할 때는 주의해서 사용해야 합니다.

② iframe 작성 방법:

1) iframe 태그를 작성

```
<iframe></iframe>
```

2) src 속성을 이용하여 삽입할 문서의 주소를 지정합니다.

```
<iframe src="삽입할문서.html"></iframe>
```

3) width와 height 속성을 이용하여 iframe의 크기를 지정합니다.

```
<iframe src="삽입할문서.html" width="500" height="300"></iframe>
```

4) frameborder 속성을 이용하여 iframe 주변에 경계선을 그릴지 여부를 지정할 수 있습니다.

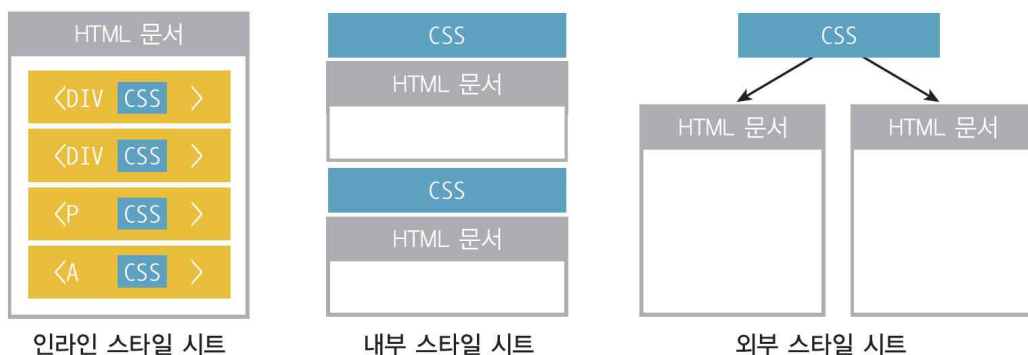
```
<iframe src="삽입할문서.html" width="500" height="300" frameborder="1"></iframe>
```

5) scrolling 속성을 이용하여 iframe 내부에 스크롤바를 표시할지 여부를 지정할 수 있습니다.

```
<iframe src="삽입할문서.html" width="500" height="300" frameborder="1" scrolling="yes"></iframe>
```

iframe은 삽입할 문서가 웹서버에 저장되어 있어야 합니다. 또한, 삽입할 문서 주소의 상대경로와 절대경로를 지정하여 경로를 지정해주어야 합니다.

3. CSS를 작성하는 3가지의 방법에 대해서 설명하고, 또한 중복작성시 우선순위에 대해서 예를 들어 설명하시오.



① 3가지 작성 방법에 대한 예시와 설명:

- 인라인:

Inline Style이라고도 합니다. 인라인 CSS 스타일의 특징은 HTML 요소의 style 속성을 이용하여 해당 요소에 직접 스타일을 작성하는 방법입니다.

```
<div style="color: red; font-size: 16px;">Hello, world!</div>
```

- 내부:

Internal Style이라는 표현을 사용하기도 합니다. 내부 CSS 스타일의 특징은 HTML 문서 내부에 <style> 태그를 이용하여 스타일을 작성하는 방법입니다.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Internal Style</title>
```

```

<style>
  div {
    color: red;
    font-size: 16px;
  }
</style>
</head>
<body>
  <div>Hello, world!</div>
</body>
</html>

```

– 외부:

External Style이라고도 합니다.

외부 CSS 스타일은 CSS 파일을 별도로 작성하여 HTML 문서에 연결하는 방법입니다.

<style.css - 외부파일>

```

div {
  color: red;
  font-size: 16px;
}

```

<html5 파일 생성 - 외부 css파일 호출>

```

<!DOCTYPE html>
<html>
  <head>
    <title>External Style</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <div>Hello, world!</div>
  </body>
</html>

```

② 중복 시 우선순위:

Inline Style > ID 선택자 > 클래스 선택자 > 속성 선택자 > 가상 클래스 선택자 > 요소 선택자 > 가상 요소 선택자 > 상위 요소에 의해 상속된 스타일 순으로 우선 순위가 적용이 됩니다.

같은 우선 순위일 경우에는 나중에 작성된 스타일이 우선순위가 됩니다.

4. 사용자가 스타일을 작성한 후, 그 스타일을 적용하기 위한 대상을 지정하기 위해서는 선택자가 필요하다. 그 중에서 주요한 3가지 선택자에 대해서 사용법 예시를 들고, 간략히 설명하여라.

① 태그 선택자

태그 선택자는 HTML에서 태그를 기반으로 스타일을 적용할 수 있는 선택자입니다.

선택하려는 HTML 태그의 이름을 그대로 작성하면 된다. 모든 p 태그의 색상을 빨간색으로 지정하려면

```
p {  
  color: red;  
}
```

를 사용하여 속성을 지정하는 것이 태그 선택자입니다.

② 클래스 선택자

클래스 선택자는 HTML 요소에 class 속성을 추가하고, 클래스 이름을 이용하여 스타일을 호출하여 적용하고 사용할 수 있습니다. 클래스 선택자는 마침표(.)로 시작하며, 선택자 이름은 임의로 정할 수 있다. 클래스 이름이 "highlight"인 요소의 색상을 노란색으로 지정하려면

```
.highlight {  
  color: yellow;  
}
```

를 사용하여 속성을 지정할 수 있는 것이 클래스 선택자입니다.

③ 아이디 선택자

아이디 선택자는 HTML 요소에 id 속성을 추가하고, id 이름을 이용하여 스타일을 호출하여 적용하고 사용할 수 있습니다. 아이디 선택자는 #으로 시작하며, 선택자 이름은 임의로 정할 수 있다. 하나의 HTML 문서에서 id 속성 값은 중복되어서는 안 되는 것이 특징입니다.

아이디 이름이 "logo"인 요소의 너비를 200픽셀로 지정하려면

```
#logo {  
  width: 200px;  
}
```

을 사용하여 속성을 지정할 수 있는 것이 id 선택자입니다.

중복 작성시 우선순위는 아이디 선택자가 가장 높고, 클래스 선택자가 그 다음이며, 태그 선택자가 가장 낮습니다. 스타일 속성에 직접 스타일을 지정한 것보다 외부 CSS 파일에 작성한 스타일이 우선시 됩니다. 또한, 스타일이 적용된 요소의 계층 구조가 복잡할수록 우선순위가 높아진다. 이러한 우선순위 규칙은 스타일 선택자가 충돌했을 때 어떤 스타일이 적용될지를 결정한다.

5. 속성 선택자의 종류 중 가상선택자는 특이하게도, 웹 문서에는 보이지 않지만 동작에 영향을 주는 속성을 선택자로 사용하는 방법이다. a(anker) 태그의 가상선택자로 예를 들어 설명하여라.

① :link

방문하지 않은 링크를 나타내는 선택자입니다.

② :visited

이미 방문한 링크에 대한 스타일을 정의하는 선택자입니다.

방문한 상태에서 다시 페이지를 로드했을 때 적용된다.

③ :active

링크를 클릭하거나 Enter 키를 눌러 활성화된 링크에 대한 스타일을 정의하는 선택자입니다.

사용자가 링크를 활성화할 때 적용이 되는 선택자입니다.

④ :hover

마우스 포인터를 올렸을 때의 상태를 나타낸다. 링크에 대한 포인터(마우스) 이벤트가 발생했을 때 적용된다.

⑤ :focus

키보드 포커스를 받았을 때의 상태를 나타낸다. 웹 접근성을 고려하여 키보드로 사이트를 탐색하는 사용자들을 위해 사용된다.

6. 여러 가지 선택자를 복합적으로 조합하는 방법에 대해서 설명하여라.

① 선택자A 선택자B

: 선택자A와 선택자B로 각각 요소를 선택한 후, 이 둘을 조합하여 선택합니다. 선택자A의 하위 요소 중 선택자B와 일치하는 모든 요소를 선택합니다.

② 선택자A > 선택자B

: 선택자A의 바로 아래 자식 요소 중에서 선택자B와 일치하는 모든 요소를 선택합니다.

③ 선택자A + 선택자B

: 선택자A + 선택자B: 선택자A의 다음 형제 요소 중에서 선택자B와 일치하는 첫 번째 요소만 선택합니다.

④ 선택자A ~ 선택자B

: 선택자A의 다음 형제 요소 중에서 선택자B와 일치하는 모든 요소를 선택합니다.

⑤ 선택자A, 선택자B

: 선택자A와 선택자B에 일치하는 모든 요소를 선택합니다.

7. 아래 예시는 transition 속성과 속성값에 대한 의사코드이다. transition 속성과 4개의 속성값에 대해서 설명하여라.

```
div { transition: property duration timing-function delay; }
```

transition이란?

웹 요소가 변화할 때 부드러운 애니메이션 효과를 적용하는 데 사용됩니다.

① transition 속성: 요소에 전체적인 트랜지션을 설정합니다.

② 1.property: 트랜지션 효과가 적용될 CSS 속성명을 지정합니다. 여러 개의 속성을 쉼표로 구분하여 지정할 수 있습니다.

2.duration: 트랜지션이 일어날 시간을 지정합니다. 속성값은 초 단위 또는 밀리초 단위가 될 수 있으며, 기본값은 0초입니다.

3.timing-function: 트랜지션 효과의 타이밍 함수를 지정합니다.

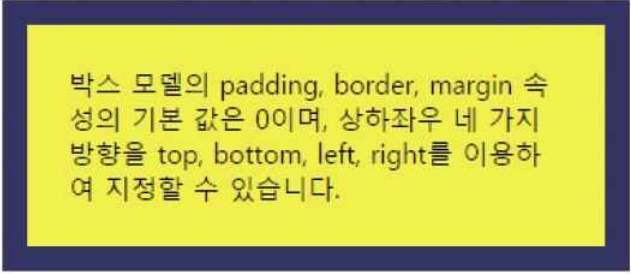
즉, 변화가 일어나는 속도를 조절하는 함수입니다. ease, linear, ease-in, ease-out, ease-in-out, cubic-bezier 등을 이용하여 timing-function의 값을 사용할 수 있습니다.

4.delay: 트랜지션 시작 전에 지연할 시간을 지정합니다.

속성값은 초 단위 또는 밀리초 단위가 될 수 있으며, 기본값은 0초입니다.

8. 아래 그림은 화면을 구성하는 개념 중 박스(box) 모델에 대한 것이다. 먼저 박스 모델에 대해서 설명한 후,

박스 모델을 구성하는 속성들에 대해서 설명하여라



박스 모델의 padding, border, margin 속성의 기본 값은 0이며, 상하좌우 네 가지 방향을 top, bottom, left, right를 이용하여 지정할 수 있습니다.

① Box 모델 개념:

HTML 요소를 박스 형태로 처리하는 모델이며, 박스 모델은 content, padding, border, margin으로 구성되어 있습니다. 박스 모델의 각 영역의 크기와 스타일을 지정하여 디자인할 수 있음

② 1.content: content는 HTML 요소의 실제 내용이 들어가는 영역입니다. content의 크기는 width, height 속성으로 지정하여 디자인을 할 수 있습니다.

2.padding: content와 border 사이의 여백 영역을 의미하며, padding의 크기는 padding-top, padding-bottom, padding-left, padding-right 속성으로 padding의 속성을 지정하며 또한 padding의 배경색을 지정하여 디자인을 할 수 있습니다.

3.margin:

border와 다른 요소 사이의 여백 영역이 margin이다. margin의 크기는 margin-top, margin-bottom, margin-left, margin-right 속성으로 margin의 속성을 지정하여 디자인한다. margin의 배경색을 지정할 수 없다.

4.border:

content와 padding을 감싸는 테두리 영역이 border입니다.

border의 두께와 스타일을 border-width, border-style 속성으로 지정하여 디자인을 할 수 있습니다.

border 색상은 border-color 속성으로 지정하여 디자인을 할 수 있습니다.

9. 화면의 레이아웃을 구성하는 위해 요소(텍스트, 이미지, 표)를 배치하여 위치를 잡기위해 사용되는 position 속성의 속성값에 대해서 예를 들어 그림과 코드로 설명하여라.

CSS의 position 속성은 화면의 레이아웃을 구성할 때 요소들을 배치하는데 사용되며, 요소의 위치를 정하는 방법에 따라 static, relative, absolute, fixed 속성값을 사용할 수 있습니다.

① static: position 속성의 기본 값으로, 화면에서 요소가 기본 위치에 배치가 됩니다. 다른 속성 값들과 달리 위치를 조정할 수 없는 것이 특징입니다.

② relative: relative는 요소의 위치를 기준으로 상대적인 위치를 지정할 때 사용 할 수 있습니다. top, bottom, left, right 속성과 함께 사용하여 요소의 위치를 이동시킬 수 있습니다.

③ absolute: absolute는 요소가 속한 부모 요소를 기준으로 상대적인 위치를 지정하지 않고, 문서 전체를 기준으로 위치를 지정할 때 사용할 수 있습니다.

부모 요소에 position 속성이 지정되어 있어야 하는 것이 특징입니다.

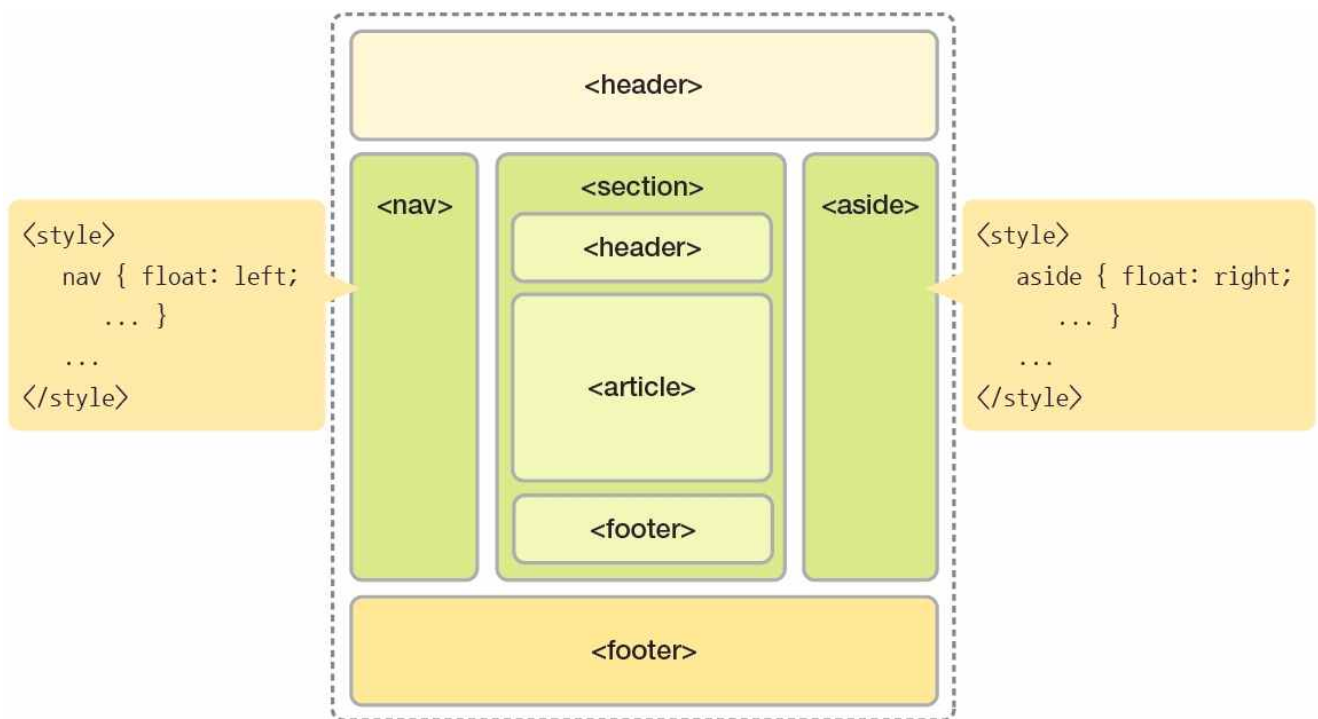
top, bottom, left, right 속성과 함께 사용하여 요소의 위치를 이동시킬 수 있습니다.

부모 요소가 없는 경우 문서 body를 기준으로 위치를 지정하게 됩니다.

④ fixed:

fixed는 문서의 스크롤을 움직여도 항상 같은 위치에 고정되어 있어야 하는 요소에 사용됩니다.
top, bottom, left, right 속성과 함께 사용하여 요소의 위치를 이동시킬 수 있습니다.

10. 콘텐츠의 내용에 의미를 부여하는 시맨틱 태그중, 다음의 3가지에 대해서 비교 설명하고, <section>, <article>, <div> 하여라. 그리고 <aside>와 <nav> 태그에 주로 사용되는 속성중, float와 clear, 그리고 overflow에 대해서 설명하여라 (20점)



-콘텐츠 시맨틱 태그

콘텐츠 시맨틱 태그는 콘텐츠의 의미를 명확하게 해주는 역할을 합니다.

웹 사이트의 가독성과 유지보수성을 높이기 위해서 콘텐츠 시맨틱 태그를 사용합니다.

① <section>

<section> 태그는 문서나 앱의 일반적인 섹션을 정의하며, 해당 섹션의 주제나 콘텐츠를 그룹화하는 역할을 합니다. <section> 태그에서는 일반적으로 제목이 있고 관련된 내용이 포함됩니다.

② <article>

<article> 태그는 문서, 페이지, 앱 내에서 독립적인 콘텐츠를 표현하는 데 사용됩니다. 주로 블로그 게시물, 뉴스 기사, 포럼 게시물 등과 같이 자체적으로 의미를 가지고 있고 다른 콘텐츠와는 독립적으로 읽을 수 있는 콘텐츠에 사용되는 편입니다.

③ <div>

<div> 태그는 단순히 콘텐츠를 그룹화하는 역할을 합니다. <div>태그는 다른 시맨틱 태그가 적용되지 않을 때 사용됩니다.

- 배치 속성

① float

float 속성은 해당 요소를 왼쪽이나 오른쪽으로 띄웁니다.

다른 콘텐츠가 그 주위를 감쌀 수 있도록 하는 속성이며, 주로 이미지나 텍스트를 감싸는 레이아웃을 구성할 때 많이 사용됩니다.

② clear

clear는 float 속성이 적용된 요소 다음에 오는 요소의 흐름을 제어하는 속성입니다.

clear 속성을 사용하면 float 속성이 적용된 요소 아래에 오는 다른 요소들이 그 요소를 감싸지 않고, 지정된 방향으로 흐르게 됩니다.

즉, clear 속성은 float 속성을 적용한 요소가 주위의 콘텐츠에 영향을 미치지 않도록 하기 위해 사용됩니다.

clear 속성의 값으로 left, right, both, none 등이 있습니다.

③ overflow

overflow는 요소의 내용이 자신의 크기를 벗어났을 때, 어떻게 처리할지를 지정하는 속성입니다. overflow 속성의 값으로는 visible, hidden, scroll, auto 등이 있습니다. visible은 자식 요소의 크기를 무시하고 부모 요소의 크기를 무시하지 않습니다. hidden은 자식 요소의 크기가 부모 요소보다 크면 자르고, 그렇지 않으면 표시하지 않습니다. scroll은 자식 요소의 크기가 부모 요소보다 크면 스크롤바를 생성합니다. auto는 내용이 부모 요소의 크기를 넘치면 스크롤바를 생성하고, 그렇지 않으면 스크롤바를 생성하지 않습니다.

aside와 nav의 시멘틱 태그

<aside>와 <nav> 태그는 주로 레이아웃에서 사용되는 시멘틱 태그입니다.

<aside>와 <nav> 태그는 주로 사이드바나 네비게이션 바 등을 표현하는 데 사용되는데, 이 때 float 속성을 적용해 레이아웃을 구성하고, clear 속성으로 주위의 콘텐츠에 영향을 받지 않도록 처리한다. overflow 속성은 해당 요소의 내용이 크기를 넘어갈 때 스크롤바를 표시하거나 숨기는 등의 처리를 할 수 있습니다.