

5장

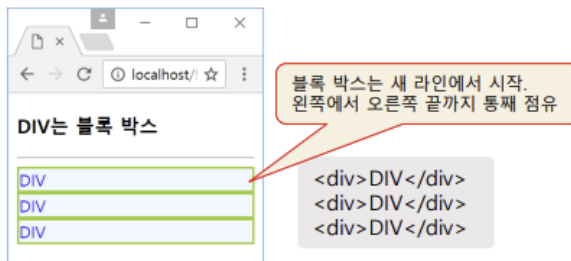
css3 고급 활용

배치

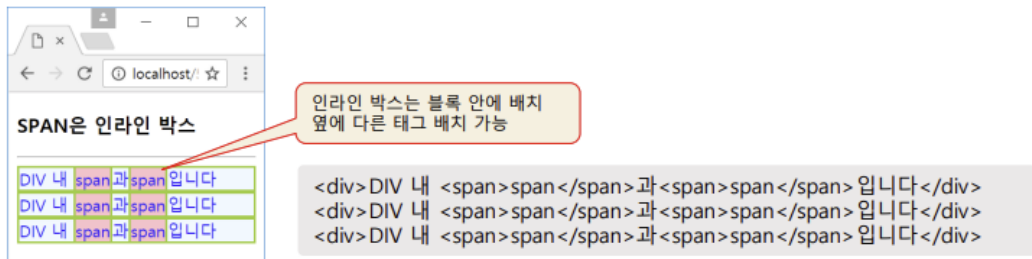
⇒ css로 HTML 태그가 출력되는 위치 지정

- 블록 박스 : 블록 태그
- 인라인 박스 : 인라인 태그

블록 박스

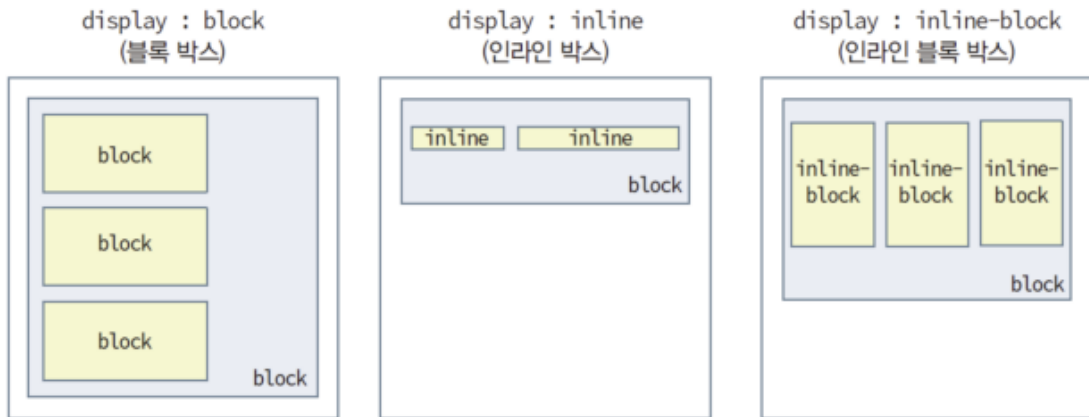


인라인 박스



- 박스의 유형 제어

: display



| 블록박스 (display:block) | 인라인 박스 (display:inline) | 인라인 블록 박스 (display:inline-block) |
|-------------------------------|------------------------------------|-------------------------------------|
| 항상 새 라인에서 시작 | 새 라인에서 시작 못함. 라인 안 (inline)에 있음 | 새 라인에서 시작 못함. 라인 안 (inline)에 있음 |
| 블록 박스 내에만 배치 | 모든 박스 내 배치 가능 | 모든 박스 내 배치 가능 |
| 옆에 다른 요소 배치 불가능 | 옆에 다른 요소 배치 가능 | 옆에 다른 요소 배치 가능 |
| width와 height으로 크기 조절 | width와 height으로 크기 조절 불가능 | width와 height 크기 조절 가능 |
| padding, border, margin 조절 가능 | margin-top, margin-bottom 조절 불가능 | padding, border, margin의 조절 가능 |

- block
- inline
- inline - block

: position

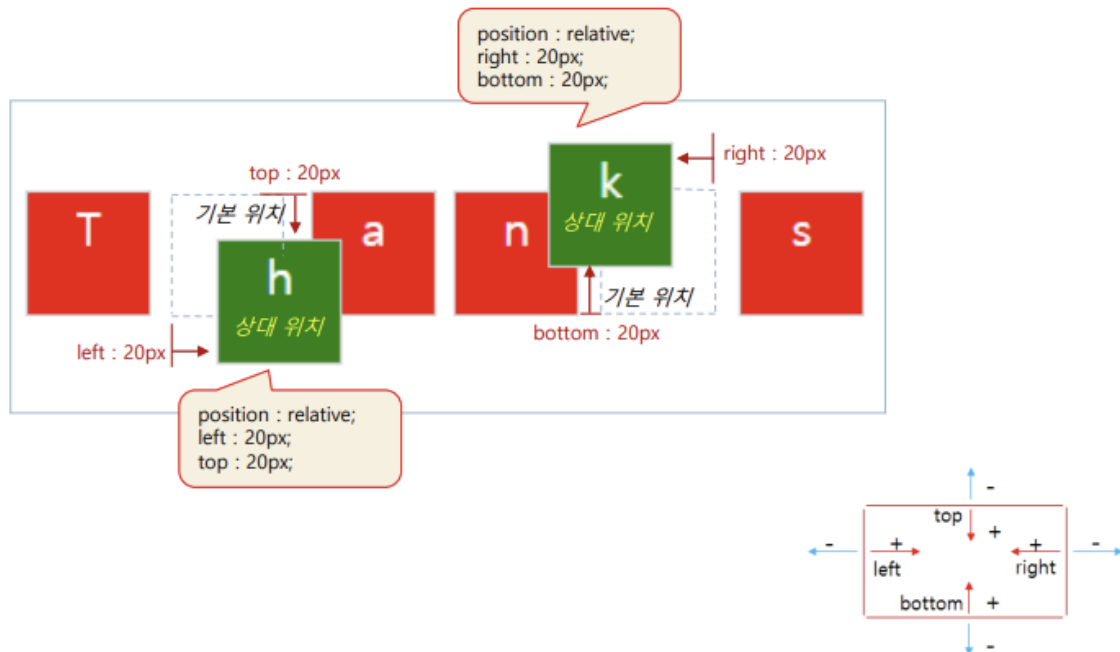
→ normal flow : 기본 위치

- 정적 배치 - position : static(디폴트)
- 상대 배치 - position : relative
- 절대 배치 - position : absolute
- 고정 배치 - position : fixed
- 유동 배치 - float : left 혹은 float : right

→ position : top, bottom, left, right, width, height 프로퍼티로 지정

1. 상대 배치 , positon : relative

⇒ normal flow의 기본 위치에서 left, top, bottom, right 프로퍼티의 값만큼 이동한 '상대 위치'에 배



2. 절대 배치 , position : absolute

⇒ 브라우저 크기가 변해도 절대 배치된 태그 위치는 변하지 않는다.

3. 고정 배치 , position : fixed

⇒ 브라우저의 정해진 위치에 고정 (브라우저 크기가 변해도 그 위치에 고정)

: float : right

⇒ 브라우저의 오른쪽에 항상 배치

: z-index

⇒ position : absolute와 관련

⇒ 요소의 쌓이는 순서(z-order)를 지정

⇒ 중첩된 요소들 간에만 작동합니다. 즉, 같은 부모 요소를 가진 요소들 간에서만 적용.

:visibility

⇒ 텍스트를 숨기는 기능

⇒ ex) visibility : hidden ;

:overflow

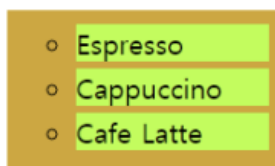
⇒ 요소 내용이 자신의 컨테이너를 넘어갈 때 처리 방법을 지정

- **visible** : 기본값으로, 요소가 컨테이너를 넘어가더라도 자르지 않고 그대로 표시합니다.
- **hidden** : 요소가 컨테이너를 넘어가면 자르고, 자른 부분을 표시하지 않습니다.
- **scroll** : 요소가 컨테이너를 넘어가면 스크롤바를 추가하여 스크롤바를 통해 자른 부분을 볼 수 있게 합니다.
- **auto** : 요소가 컨테이너를 넘어가면 스크롤바를 추가하여 스크롤바가 필요한 경우에만 자른 부분을 볼 수 있게 합니다. 스크롤바가 필요하지 않은 경우 자르지 않습니다.

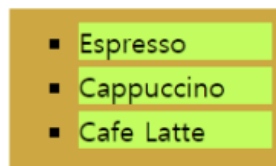
리스트 꾸미기

- list-style-type : 아이템 마커 타입 지정

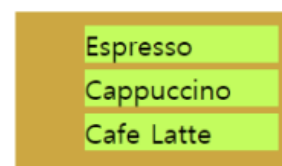
```
list-style-type : disc | armenian | circle | cjk-ideographic | decimal | georgian | lower-alpha | lower-roman | square | upper-alpha | upper-roman | none
```



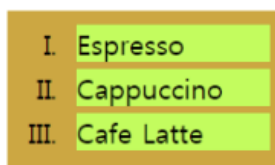
(a) list-style-type : circle



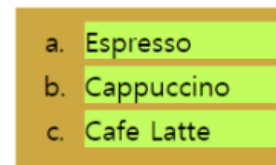
(b) list-style-type : square



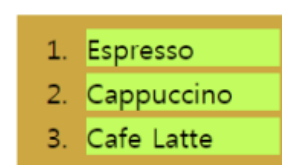
(c) list-style-type : none



(d) list-style-type : upper-roman



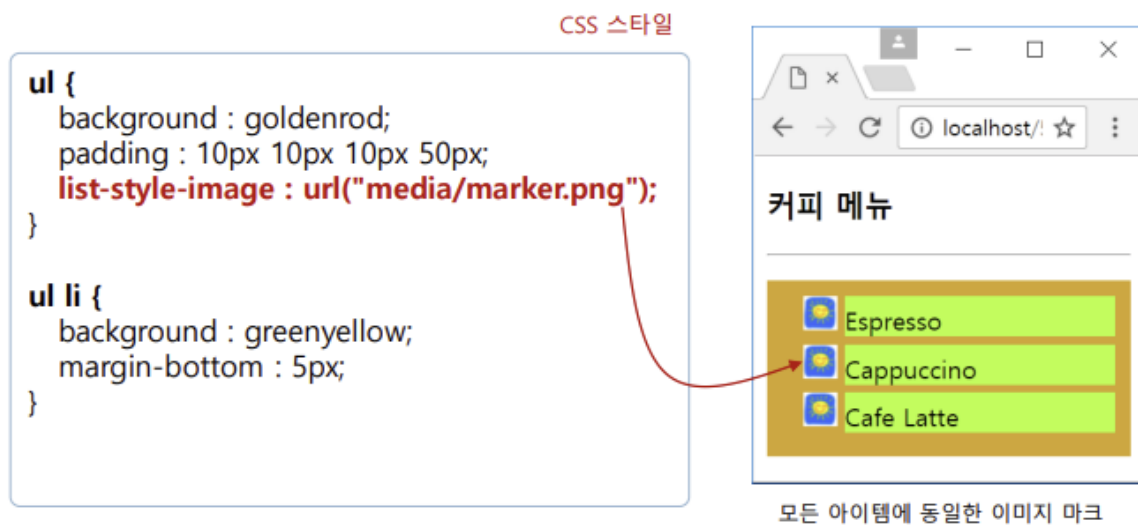
(e) list-style-type : lower-alpha



(f) list-style-type : decimal

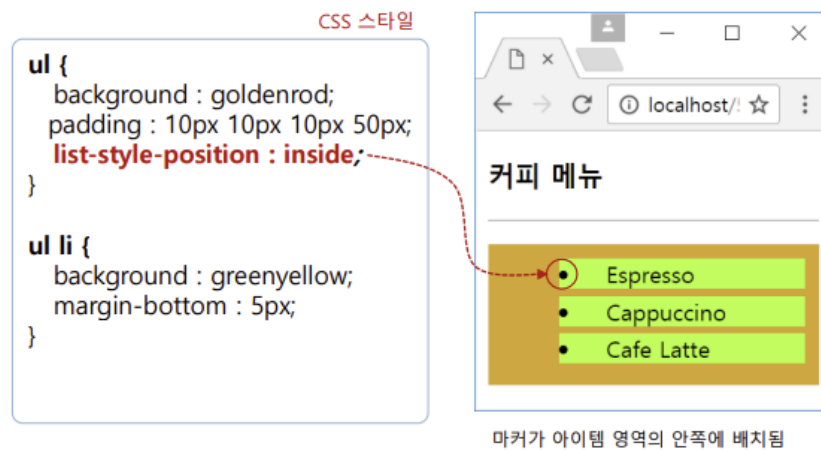
- list-style-image : 아이템 마커 이미지 지정

→ 사용자가 이미지 마커 작성 가능



- list-style-position : 아이템 마커의 출력 위치 지정

`list-style-position : inside|outside`



- list-style : 앞의 3개 프로퍼티 값을 한 번에 지정하는 단축 프로퍼티

표 꾸미기

: CSS3로 표 꾸미기를 설명할 기본 HTML 문서

: border - 표 테두리 제어

⇒ border-collapse → 중복된 테두리 합치기
: width, height - 셀 크기 제어
: 셀 여백 - padding
: 셀 정렬 - text-align
: 배경 색 - background
: 테두리 - border -top,bottom
: 줄무늬 만들기 - 짝수 or 홀수 행의 배경색 지정
⇒ tbody tr : nth-child(even)
⇒ tbody tr : nth-child(odd)

폼 꾸미기

- input [type = text]

⇒ 글자 색 지정 → color : red
⇒ 테두리 만들기 → border : ~
⇒ 포커스를 받을 때 ; focus { }

동적 변화

1. 애니메이션

→ @keyframes

```
@keyframes name {  
  시간비율 { 스타일; 스타일; } /* 시간비율 시점까지 적용할 스타일 시트 작성 */  
  .....  
  시간비율 { 스타일; 스타일; } /* 이전 시점에서 시간비율 시점까지 적용할 스타일 시트 작성 */  
}
```

```
@keyframes textColorAnimation {  
  0% { color : blue; } /* 시작 시. 0% 대신 from 사용 가능 */  
  30% { color : green; } /* 30% 경과 시까지 */  
  100% { color : red; } /* 끝까지. 100% 대신 to 사용 가능 */  
}
```

→ 애니메이션 스타일 시트

```

animation-name : 애니메이션 이름;           /* @keyframes의 name 지정 */
animation-duration : 시간;                   /* 1회 애니메이션 시간 */
animation-iteration-count : 애니메이션 반복 횟수; /* 숫자로도 줄 수 있지만, infinite로 주면
                                                    무한 반복 */

```

```

span {
  animation-name : textColorAnimation; /* 애니메이션 코드 이름 */
  animation-duration : 5s;             /* 애니메이션 1회 시간은 5초 */
  animation-iteration-count : infinite; /* 무한 반복 */
}

```

2. 전환 (transition)

⇒ HTML 태그에 적용된 CSS 프로퍼티 값의 변화를 **서서히** 진행시켜 애니메이션 효과 생성

⇒ 전환 설정 : transition 프로퍼티

 태그의 글자 크기에 변화가 일어나면, 5초에 걸쳐 서서히 변화가 진행되도록 전환 지시

 태그에 마우스를 올리면 글자 크기를 500% 확대 -> 전환 효과시작

CSS 스타일

```

span {
  transition : font-size 5s;
}
span:hover {
  font-size : 500%;
}

```

HTML 코드

```

<span> 팡!</span>

```

3. 변환(transform)

⇒ 텍스트나 이미지를 회전, 확대 다양한 기하학적인 모양으로 출력

⇒ 회전 각도의 단위는 deg (시계 방향의 회전)

<2차원 변환 함수>

| 변환 함수 | | 설명 |
|-------|------------------------|--|
| 위치 이동 | translate(x,y) | 태그를 X-축, Y-축으로 x, y 만큼 이동 |
| | translateX(n) | 태그를 X-축으로 n 만큼 이동 |
| | translateY(n) | 태그를 Y-축으로 n 만큼 이동 |
| 확대/축소 | scale(w,h) | 태그의 폭과 높이를 각각 w, h 배 만큼 조절. w나 h를 0으로 주면 보이지 않게 됨 |
| | scaleX(n) | 태그의 폭을 n배 만큼 조절 |
| | scaleY(n) | 태그의 높이를 n배 만큼 조절 |
| 회전 | rotate(angle) | 태그를 angle 각도 만큼 시계 방향 회전 |
| 기울임 | skew(x-angle, y-angle) | 태그를 X-축과 Y-축을 기준으로 각각 x-angle, y-angle 각도만큼 기울임 변환 |
| | skewX(angle) | 태그를 X-축을 기준으로 angle 각도만큼 기울임 |
| | skewY(angle) | 태그를 Y-축을 기준으로 angle 각도만큼 기울임 |