Loves UX, frontend & buamdong

FLOAT
& POSITION

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

왜 CSS가 어려울까?

CSS가 브라우저에서 어떻게 렌더링되는 지를 이해하지 않고 무작정 쓸려고 했기 때문!

CSS도 나름대로 원리가 다 있어요.
우리의 목표는 **물아일체의 단계**로 도약하는 것!

- 1단계 : 어머나ㅎㅎ 레이아웃이 다 박살났네 ㅎㅎ
- 2단계 : 뭔가 되었긴 한데 나도 어떻게 됐는진 모름ㅋㅋㅋㅋ
- 3단계 : ㅎ이건 이렇게 될 수 밖에 없어!!!

Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

오늘 함께할 새로운 CSS 이야기

Float, 어렵지 않아요!

레이아웃 파사삭의 주범, Float의 개념을 천천히 짚어봅시당

과제, 이젠 어렵지 않아요!

- 상단 네비게이션 바
- 3단 컬럼 디자인
- 과제 질의 응답

+ 언제 Float를 쓰고 언제 Position을 써야할지 넘나 헷갈리는 것!

Position, 어렵지 않아요!

레이아웃 파사삭의 또다른 주인공, Position의 개념도 짚어보아용

Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

블록의 본질

길막

Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

Loves UX, frontend & buamdong

CSS
블록 요소
기본값 복습

Default computed style:
block

block의 본질은 블록킹 = 길막,
빈 공간은 **`margin`**이 자동으로 채워져요.

별도의 **`width`** 값을 지정하지 않으면,
부모의 **`content-box`**의 길이 만큼 **`width`**값이 설정!

별도의 height 값을 설정하지 않으면,
자식 요소의 크기만큼 height 값이 설정!

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

CSS에서 인라인은 도대체 뭔데?

```
<p>
 Lorem ipsum dolor sit amet, consectetur adipisicing
 elit, sed do eiusmod tempor incididunt ut labore et
 dolore magna aliqua. Ut enim ad minim veniam, quis
 nostrud exercitation ullamco laboris nisi ut aliquip
 ex ea commodo consequat. Duis aute irure dolor in
 reprehenderit in voluptate velit esse cillum dolore eu
 fugiat nulla pariatur. Excepteur sint occaecat
 cupidatat non proident, sunt in culpa qui officia
 deserunt mollit anim id est laborum.
</p>
```

Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

흐름을 타는 인라인

블록 : 인라인 = 면 : 선
블록 : 인라인 = 영역 : 흐름

```
width, height,
padding-top, padding-bottom,
margin-top, margin-bottom,
```

이런 값들은 오로지 '영역'을 차지할 수 있는 블록 요소에게만 적용 가능합니당

Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

Float

Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

가로배치

Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

플로트의 기본 동작 원리

Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

자식, 어디로 갔나요?

`float`의 특징 #1

이제부터 넌 집나간 자식

float
ON WATER/IN AIR | [자동사][V + adv. / prep.]
(물 위나 공중에서) 떠[흘러]가다**[떠돌다]**

`float: none`	부모
`float: left`	부모 ;;;;;;

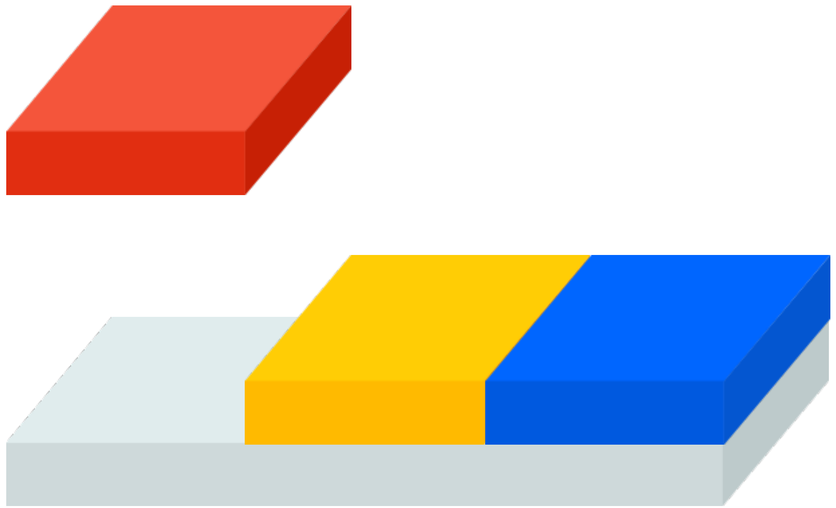
/* 이를 **Normal Flow에서 벗어난다**라고 말할 수도 있습니다. */

Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

플로트의 기본 동작 원리



Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

자동
신분 상승

`float`의 특징 #2

float이 되는 순간 자동 신분 상승,
너는 이제 `block`

`width, height, padding, margin` 값 설정 가능!

Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

길막은 이제 그만~

`float`의 특징 #3

별도의 값을 주지 않으면 부모의 `content-box` 만큼 늘어났던 `width`도 굳바이!

콘텐츠 크기가 아무리 작아도, 별도로 width값을 주지 않으면 부모의 content-box 사이즈만큼 영역을 차지 했던 녀석이
이제는 자신의 콘텐츠 크기 만큼 값을 가집니다.

실제 콘텐츠 크기

실제 콘텐츠 크기

```
float: left
```

/* 두 예제 모두 별도의 `width`값을 안 준 상태 */

JISEUNG ROH
jiseungroh@likelion.org

길막은 이제 그만~

`float`의 특징 #3

블록 요소라면 자동으로 채워지는 `margin`은 굳바이!

자동으로 채워지는 margin이 사라지면,
자동으로 채워지는 margin이 사라져야만!!!
빈 공간이 생깁니다!

`float: none`

`float: left`

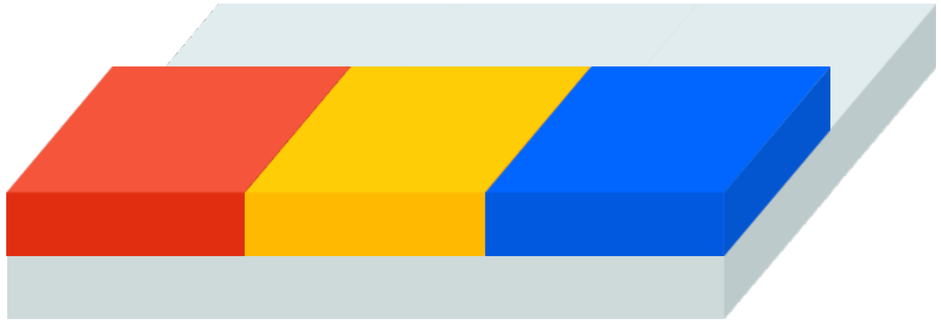
`float: right`

Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

플로트의 기본 동작 원리

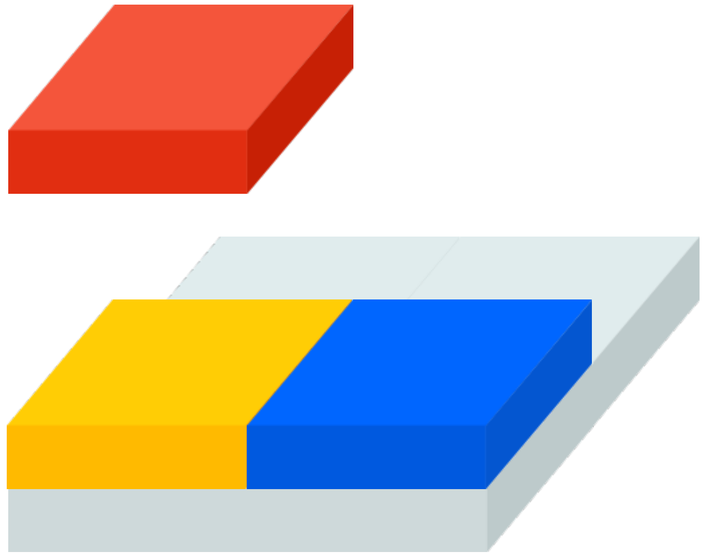


Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

플로트의 기본 동작 원리



Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

플로트 있어요.

`float`의 특징 #4

`inline`요소가 부릅니다,
“나만 볼 수 있어요~ 내 눈에만 보여요!”

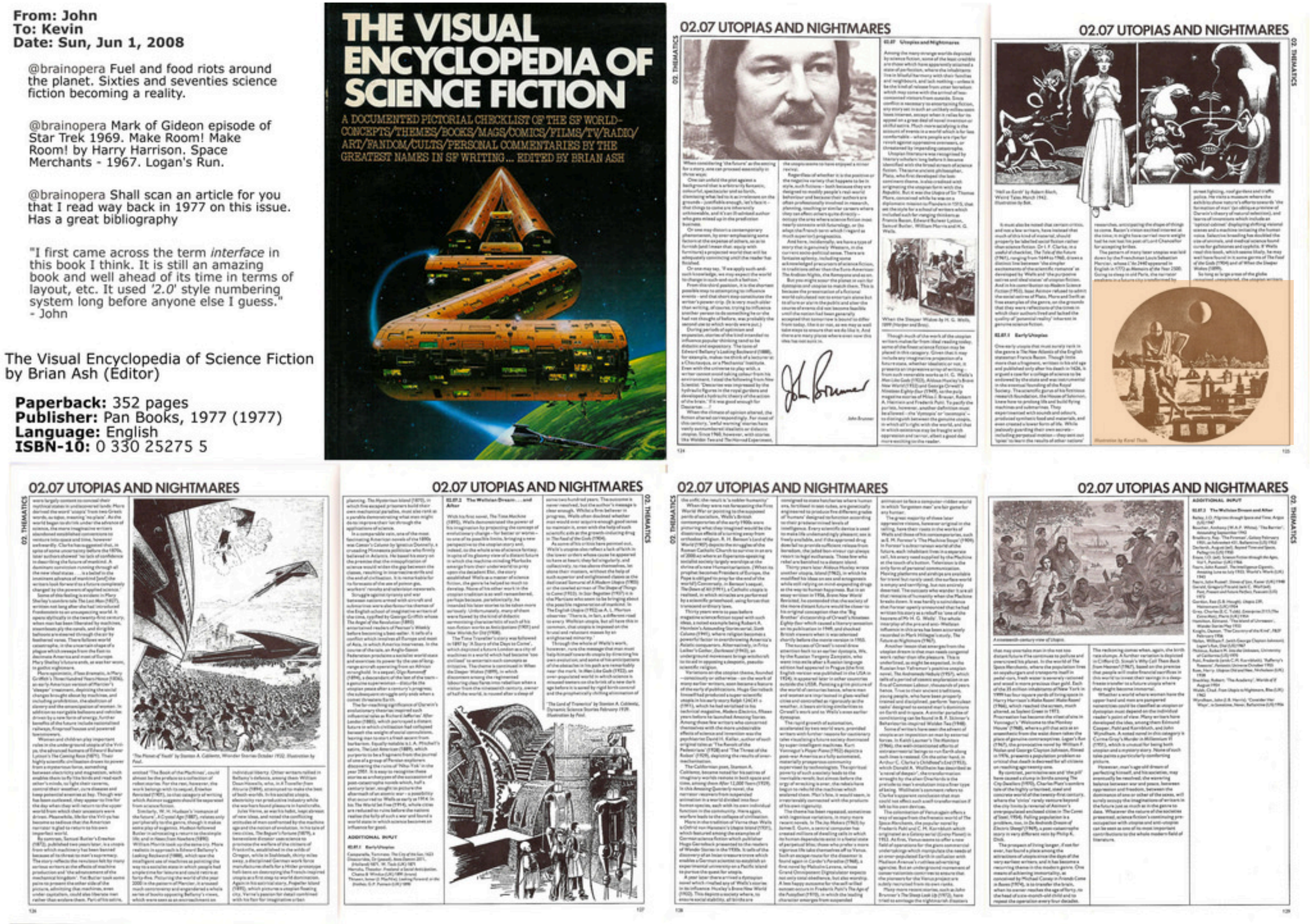
집나간 자식은 부모도 그 뒤에 따라오는 형제요소도 볼 수 없어요.
하지만 유일하게 볼 수 있는 녀석이 하나 있죠.
그리고 그것은 바로 `inline` 콘텐츠들!

왜 그러냐구요?
`float`가 만들어진 배경을 알면 이해가 쉬워요!

Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org



데헷..
그림 넣고 싶었쪙..

Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

Loves UX, frontend & buamdong

가운데정렬..? 빠른 포기가 정답

`float`의 특징 #5

그리고 딱 그 수준으로 만들어진 기능, `float`로는 `left`나 `right`밖에 적용할 수 없습니다.

float로 가운데 배치를 하시겠다구요..? ㅎㅎㅎ
당신의 정신건강을 해치는 일입니다.
빠른 포기가, 정답!

- 못하는 건 아닙니다..
- 다만 굳이.. 왜..

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

아무도 그 이유를 알지 못하는,

첫번째 방법

한번쯤 들어봤을 `overflow: hidden`

이게 왜 `float` 문제를 해결하는지 아시는 분은 갠톡ㄱㄱ

묻지도 않고, 따지지도 않고, 그냥 쓰는
`overflow: hidden`

`float: left`	부모 `{ overflow: hidden}`

Loves UX, frontend & buamdong

github.io/rohjs

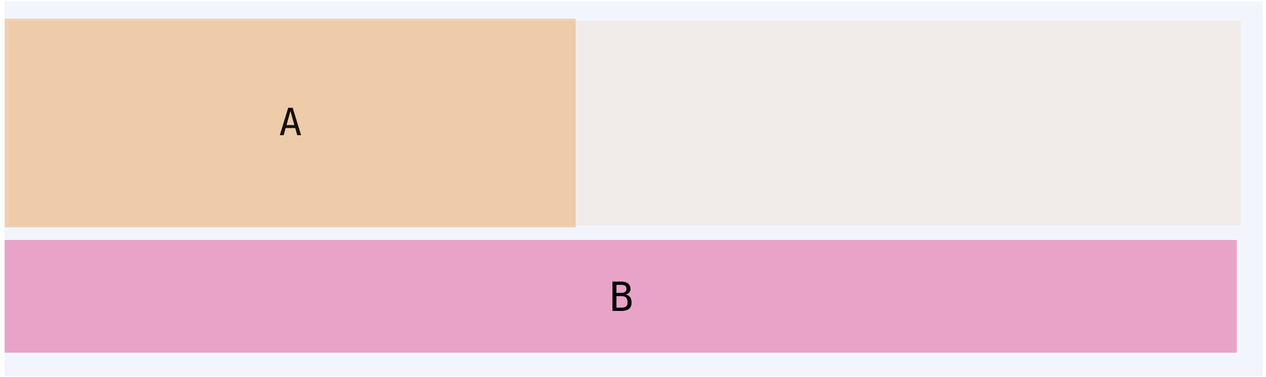
JISEUNG ROH
jiseungroh@likelion.org

플로트 해제의 정석

두번째 방법: `clearfix`

난 이제 `float`의 영향을 받지 않을 테야:
`clear: left / right / both`

`float`된 요소 다음에 오는 요소에게
`clear`속성을 주어 `float`의 영향을 받지 않게 만드는 방법입니다!



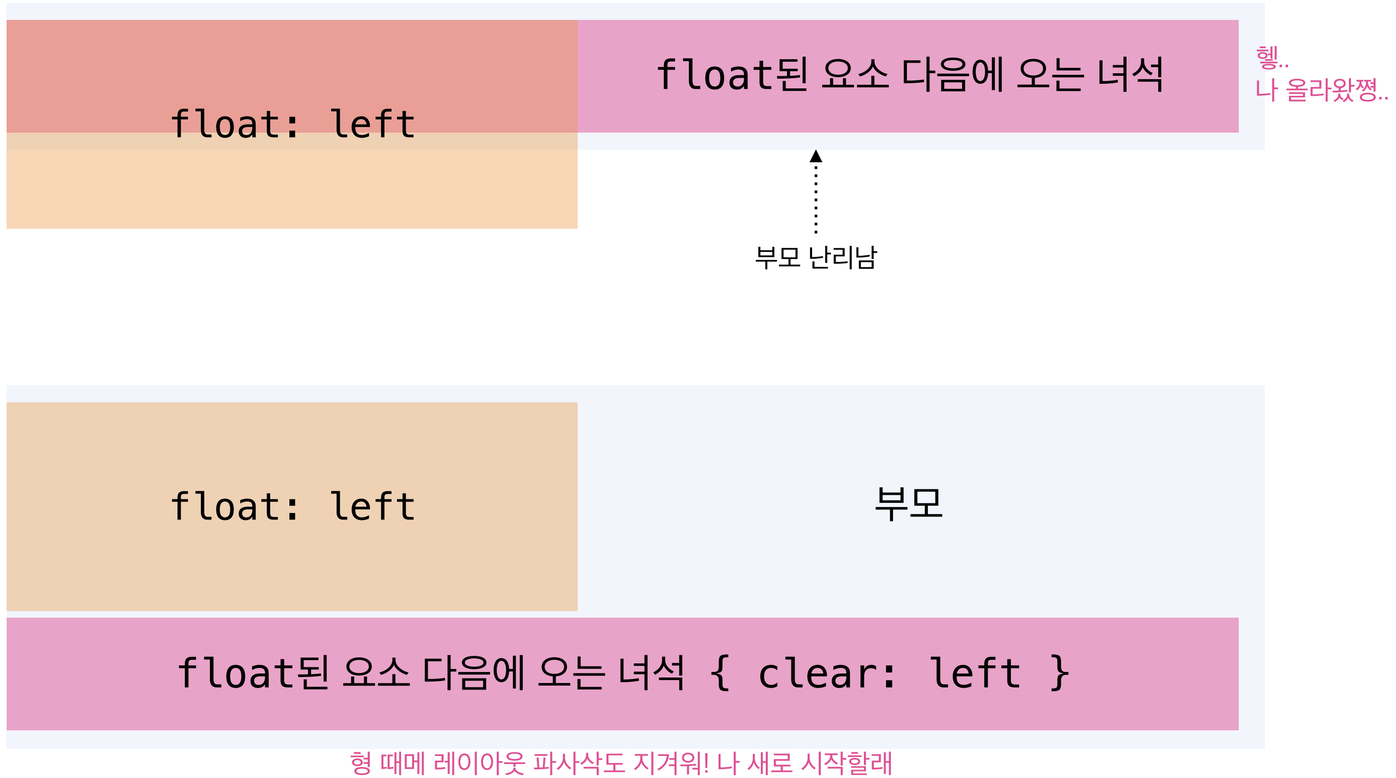
Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

플로트 해제의 정석

두번째 방법: `clearfix`



Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

플로트 해제의 정석

두번째 방법 + 가상요소 스킬


```
.parent::after {
  content: ‘’;
  display: block;
  clear: left;
}
```

=

/* .parent란 클래스 끝에 가상요소를 하나 만들건데,
content 내용은 아무것도 없고
블록 요소이고, 이 자식은
float: left의 영향을 받지 않을 거야 */

Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

tl; dr;

요약

이 클래스를 부모에게 적용시키세유

```
.clearfix::after {
  content: ‘’;
  display: block;
  clear: left;
}
```

Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

신나는 과제 풀이

네, 그렇습니다.
출제자는 접니다 ㅎㅎ

과제, 이젠 어렵지 않아요!

- 상단 네비게이션 바
- 3단 컬럼 디자인
- **과제 질의 응답**

Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

Loves UX, frontend & buamdong

Position

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

Position Static

모든 요소들의 기본 position값은 static

position값이 없는 게 아니예요.
기본값으로 position: static이 지정되어 있습니당

Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

Position Relative

상당히 상냥한 녀석

자신의 원래 위치를 기준으로 이동!
일반 흐름을 깨지 않아요!

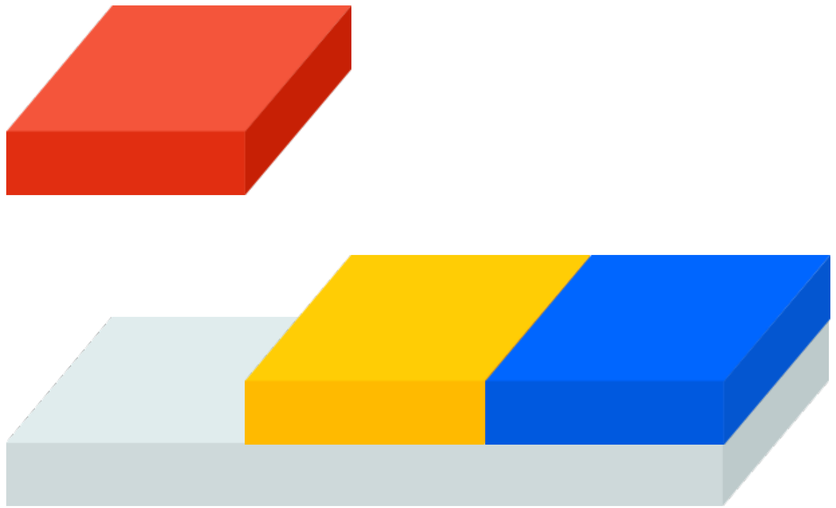
Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

Position Absolute

`float`가 집나간 자식이라면
`absolute`는 버린 자식이랄까



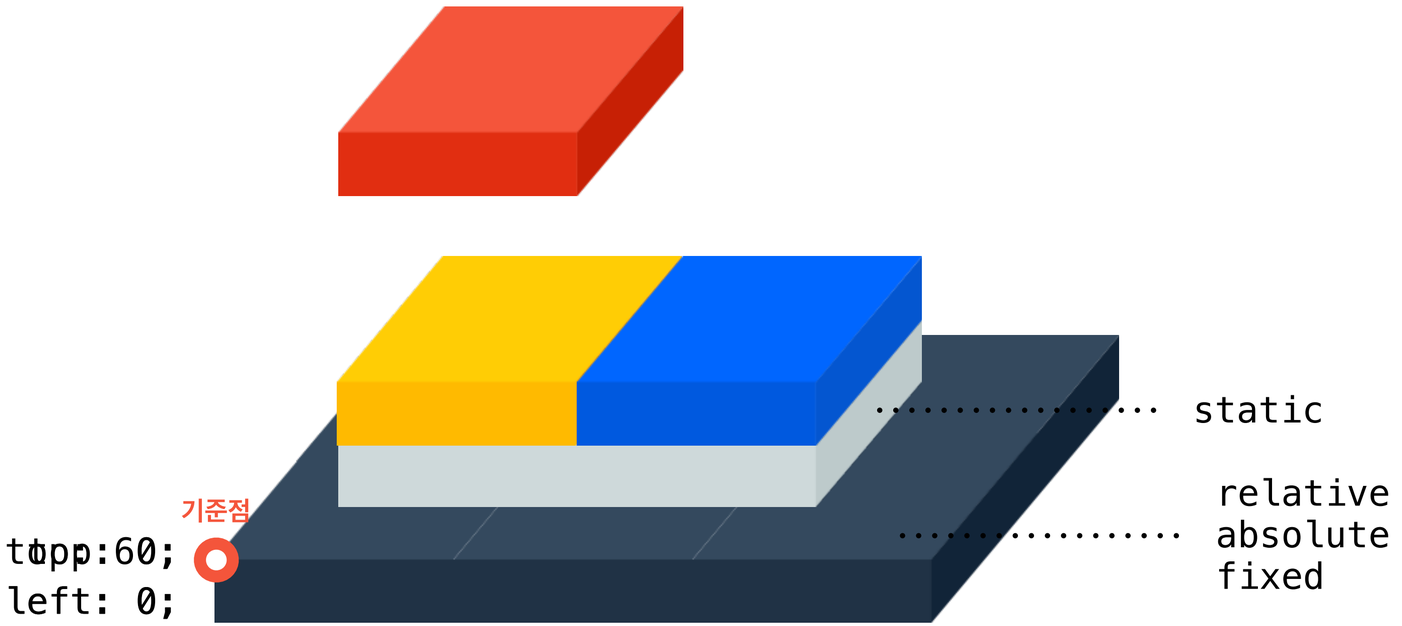
Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

Position Absolute

부모까지 바꿔버리는 무시무시한 녀석, `absolute`



Loves UX, frontend & buamdong

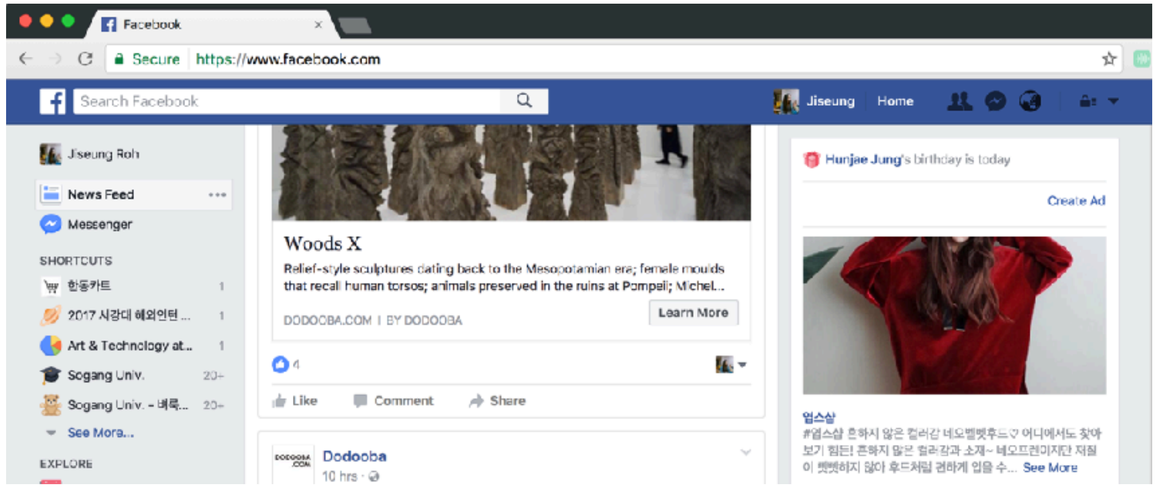
github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

Position Fixed

나의 부모는 `viewport`다!

absolute랑 똑같은, 부모가 viewport일뿐.
여기서 viewport는 브라우저 윈도우를 말합니다.
그래서 화면에서 계속 고정된 위치에 보이는 것이죠.



Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

Offset

Loves UX, frontend & buamdong

github.io/rohjs

JISEUNG ROH
jiseungroh@likelion.org

다음
이 시간에

궁금한게 있으시면, 언제든지 hit me up

jiseungroh@likelion.org
(+82) 10 9226 9306
Slack DM @jiseung_roh

Loves UX, frontend & buamdong

github.io/rohjs