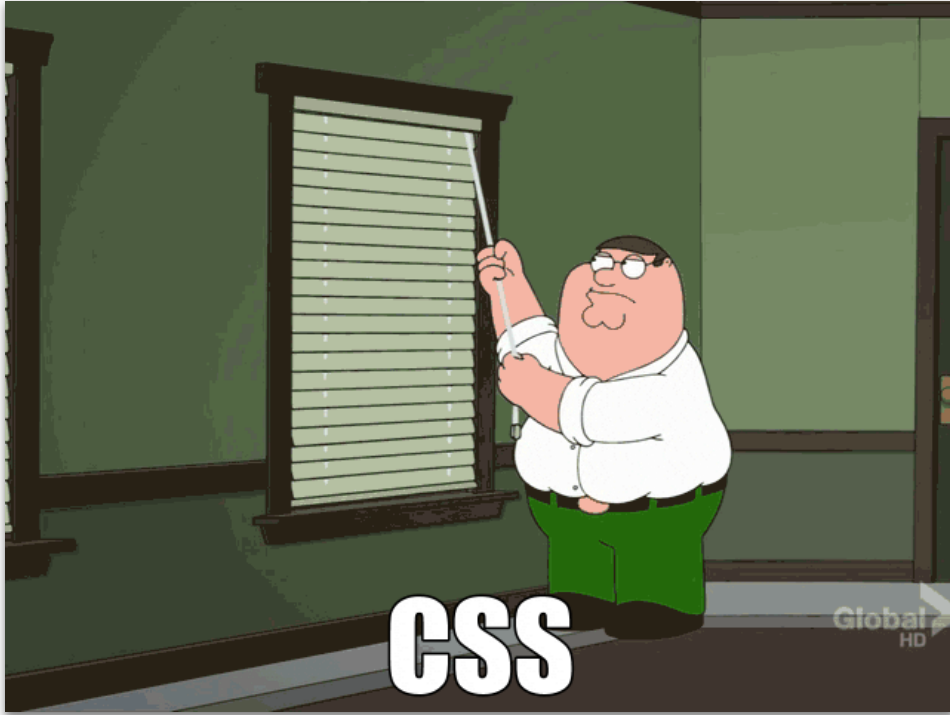


07. How to study CSS

CSS가 어려운 이유.



CSS가 어려운 이유



시행 착오에 의존하고 요행을 바라기 때문.
문제를 해결했지만 원인을 모름.

CSS에 대한 오해



“배우기 쉽다.”
“심플하다.”

배우기 쉽다. 심플하다.

Selector

p

{

color: red;

}

Property

Property value

Declaration

배우기 쉽다? 심플하다?

css float hack 🤖

🔍 전체

🖼️ 이미지

🎥 동영상

검색결과 약 4,560,000개 (0.45초)

배우기 쉽다? 심플하다?

1	Introduction
2	Background and Motivation
3	Design Goals
4	Overview
5	Grid Layout Concepts and Terminology
6	Grid Layout and Accessibility
7	Defining the Grid
8	Grid Layout and Accessibility
9	Grid Layout and Accessibility
10	Grid Layout and Accessibility
11	Grid Layout and Accessibility
12	Grid Layout and Accessibility
13	Grid Layout and Accessibility
14	Grid Layout and Accessibility
15	Grid Layout and Accessibility
16	Grid Layout and Accessibility
17	Grid Layout and Accessibility
18	Grid Layout and Accessibility
19	Grid Layout and Accessibility
20	Grid Layout and Accessibility
21	Grid Layout and Accessibility
22	Grid Layout and Accessibility
23	Grid Layout and Accessibility
24	Grid Layout and Accessibility
25	Grid Layout and Accessibility
26	Grid Layout and Accessibility
27	Grid Layout and Accessibility
28	Grid Layout and Accessibility
29	Grid Layout and Accessibility
30	Grid Layout and Accessibility
31	Grid Layout and Accessibility
32	Grid Layout and Accessibility
33	Grid Layout and Accessibility
34	Grid Layout and Accessibility
35	Grid Layout and Accessibility
36	Grid Layout and Accessibility
37	Grid Layout and Accessibility
38	Grid Layout and Accessibility
39	Grid Layout and Accessibility
40	Grid Layout and Accessibility
41	Grid Layout and Accessibility
42	Grid Layout and Accessibility
43	Grid Layout and Accessibility
44	Grid Layout and Accessibility
45	Grid Layout and Accessibility
46	Grid Layout and Accessibility
47	Grid Layout and Accessibility
48	Grid Layout and Accessibility
49	Grid Layout and Accessibility
50	Grid Layout and Accessibility
51	Grid Layout and Accessibility
52	Grid Layout and Accessibility
53	Grid Layout and Accessibility
54	Grid Layout and Accessibility
55	Grid Layout and Accessibility
56	Grid Layout and Accessibility
57	Grid Layout and Accessibility
58	Grid Layout and Accessibility
59	Grid Layout and Accessibility
60	Grid Layout and Accessibility
61	Grid Layout and Accessibility
62	Grid Layout and Accessibility
63	Grid Layout and Accessibility
64	Grid Layout and Accessibility
65	Grid Layout and Accessibility
66	Grid Layout and Accessibility
67	Grid Layout and Accessibility
68	Grid Layout and Accessibility
69	Grid Layout and Accessibility
70	Grid Layout and Accessibility
71	Grid Layout and Accessibility
72	Grid Layout and Accessibility
73	Grid Layout and Accessibility
74	Grid Layout and Accessibility
75	Grid Layout and Accessibility
76	Grid Layout and Accessibility
77	Grid Layout and Accessibility
78	Grid Layout and Accessibility
79	Grid Layout and Accessibility
80	Grid Layout and Accessibility
81	Grid Layout and Accessibility
82	Grid Layout and Accessibility
83	Grid Layout and Accessibility
84	Grid Layout and Accessibility
85	Grid Layout and Accessibility
86	Grid Layout and Accessibility
87	Grid Layout and Accessibility
88	Grid Layout and Accessibility
89	Grid Layout and Accessibility
90	Grid Layout and Accessibility
91	Grid Layout and Accessibility
92	Grid Layout and Accessibility
93	Grid Layout and Accessibility
94	Grid Layout and Accessibility
95	Grid Layout and Accessibility
96	Grid Layout and Accessibility
97	Grid Layout and Accessibility
98	Grid Layout and Accessibility
99	Grid Layout and Accessibility
100	Grid Layout and Accessibility

§ 7.8. Grid Definition Shorthand: the 'grid' property

Name: 'grid'



Value: <grid-template> | <grid-template-rows> / [auto-flow && dense?] <grid-auto-columns>?
| [auto-flow && dense?] <grid-auto-rows>? / <grid-template-columns>

Initial:

Name:

'grid-template'



Applies to:

Value:

none | [<grid-template-rows> / <grid-template-columns>] |
[<line-names>? <string> <track-size>? <line-names>?]+ [/ <explicit-track-list>]?

Inherited:

see individual properties

Percentages:

see inc

Name:

'grid-template-columns', 'grid-template-rows'



Value:

none | <track-list> | <auto-track-list>

Computed value:

see inc

Animation type:

see individual

Canonical order:

per grammar

```

<track-list>      = [ <line-names>? [ <track-size> | <track-repeat> ] ]+ <line-names>?
<auto-track-list> = [ <line-names>? [ <fixed-size> | <fixed-repeat> ] ]* <line-names>? <auto-repeat>
<explicit-track-list> = [ <line-names>? [ <fixed-size> | <fixed-repeat> ] ]* <line-names>?

<track-size>      = <track-breadth> | minmax( <inflexible-breadth> , <track-breadth> ) | fit-content( <length-percentage> )
<fixed-size>      = <fixed-breadth> | minmax( <fixed-breadth> , <track-breadth> ) | minmax( <inflexible-breadth> , <fixed-breadth> )
<track-breadth>   = <length-percentage> | <flex> | min-content | max-content | auto
<inflexible-breadth> = <length-percentage> | min-content | max-content | auto
<fixed-breadth>   = <length-percentage>
<line-names>      = '[' <custom-ident> '*' ]'
```



CSS에 대한 오해

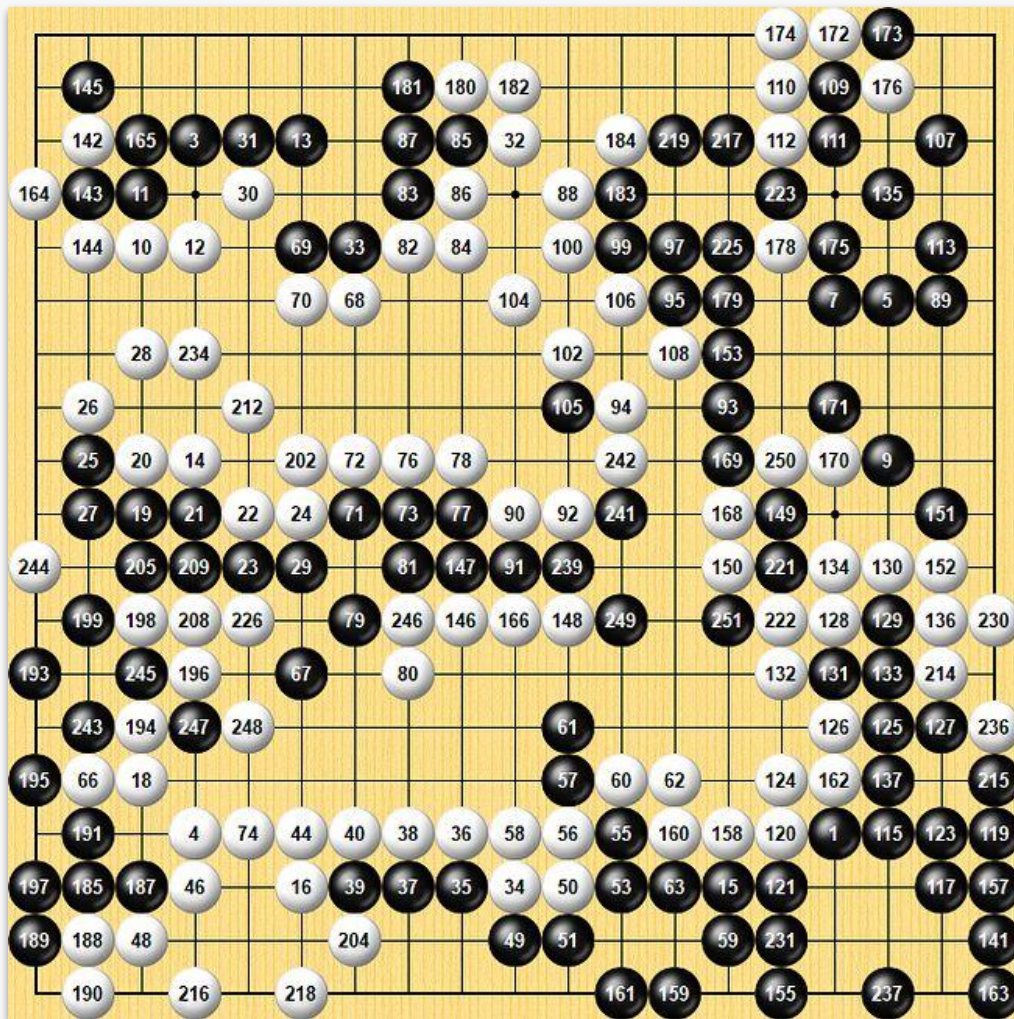
2019) CSS는 배우기 쉽습니까?

- 매우 그렇다 = 24.7%
- 그렇지 않다 = 10.1%

2020) CSS는 배우기 쉽습니까?

- 매우 그렇다 = 10.5%
- 그렇지 않다 = 19.5%

CSS를 어떻게 공부해야 하는가?



출력 결과를 분석하고 실수를 복기.

✓ CSS 명세

“

CSS Values Basic



Value definition syntax

"값 정의 구문"

CSS 속성의 유효한 값과 순서.

Property:value definition

Name:	<i>font-size</i>
Value:	<u><absolute-size></u> <u><relative-size></u> <length-percentage>
Initial:	medium
Applies to:	all elements
Inherited:	yes
Percentages:	refer to parent element's font size
Computed value:	absolute length
Animatable:	as <u>length</u>

Property:value definition ☆

- Name: 속성 이름.
- Value: 유효한 값과 순서.
- Initial: 초깃값. 상속받기 전 본래 값.
- Applies to: 적용할 수 있는 요소 또는 박스 유형.
- Inherited: 부모로부터 값을 상속 받는지 여부.

Property:value definition ☆

Name: ***'grid'***

Value: <'grid-template'> | <'grid-template-rows'> / [auto-flow && dense?] <'grid-auto-columns'>?
| [auto-flow && dense?] <'grid-auto-rows'>? / <'grid-template-columns'>

“

Values Types



유효한 값 형식 : keywords

예약된 단어.

Ex) initial, inherit, unset, block, inline, inline-block,
auto, disc, collapse, separate...

따옴표 없이 사용. 대소문자 구별 안 함.

유효한 값 형식 : <*> 기본 자료형

<string>	// 문자열. 따옴표(", ")로 구성.★
<url>	// url() 함수 ★
<integer>	// 정수 ★
<number>	// 실수 ★
<percentage>	// 백분율(%) ★
<length>	// 길이 ★
<position>	// 위치(left, center, right, top, bottom, px, %...) ★
<color>	// 색상 ★
<image>	// 이미지(<url> <gradient>) ★
<angle>	// 각도(deg ★, ...)
<time>	// 시간(s ★, ms)
<custom-ident>	// 작성자 정의 임의 식별자. 대소문자 구별. ★
<ratio>	// 비율(너비/높이)
<frequency>	// 빈도(Hz, kHz)
<resolution>	// 해상력(dpi, dpcm, dppx)

유효한 값 형식 : <*> vs <'*>

<따옴표>가 없으면 기본 자료형,
<'따옴표'>가 있으면 그 이름의 속성 값을 참조.

Ex) <absolute-size>

기본 자료형 [xx-small | x-small | small | medium | large | x-large | xx-large]

Ex) <'grid-template'>

grid-template 속성의 value 형식을 참조. 보통 단축 속성 값 정의 구문에 등장.

“

Combinators / Multipliers



Combinators / Multipliers

§ 7.8. Grid Definition Shorthand: the `'grid'` property

Name: `'grid'`

Value:

`<'grid-template'> | <'grid-template-rows'> / [auto-flow && dense?] <'grid-auto-columns'>?
| [auto-flow && dense?] <'grid-auto-rows'>? / <'grid-template-columns'>`

Initial: none

Applies to: `grid containers`

구성 요소 값 결합 기호 (Component Value Combinators)

구성 요소 값 증가 기호 (Component Value Multipliers)

Combinators (결합 기호)

1. 공백(^{and}): 둘 이상의 값이 필수. 순서 유지 필수.
2. &&(^{and}): 둘 이상의 값이 필수. 순서 변경 가능.
3. ||(^{or}): 두 값 중 하나 이상 필수. 순서 변경 가능.
4. |(^{only}): 두 값 중 하나만. ☆
5. [](^{group}): 그룹.

1~4는 우선 순위. 우선 순위가 높은 기호를 먼저 해석해야 한다.

Combinators (결합 기호)

$a \ b \mid c \ d \leftarrow ?$

$a \ [b \mid c] \ d \leftarrow \times$

$[a \ b] \mid [c \ d] \leftarrow \circ$

Combinators (결합 기호)

a b | c || d && e f ← ?

Combinators (결합 기호)

a b | c || d && e f
[a b] | c || d && [e f]
[a b] | c || [d && [e f]]
[a b] | [c || [d && [e f]]]

우선 순위에 따라 '공백 → && → || → |' 순서로
그룹핑하면서 해석하면 된다.

Multipliers (증가 기호)

1. *: 횟수 제한 없음. $0 \sim \infty$ 가능.
2. +: 1회 이상.
3. ?: 0회 또는 1회.
4. {A}: 정확히 A회.
5. {A, B}: 최소 A회, 최대 B회.
6. {A,}: 최소 A회 필요, 최댓값 무제한.
7. #: 1회 이상. 값을 콤마(,)로 분리. 횟수 제한 가능. 예) `<length>#{1, 4}`
8. []!: 그룹에서 적어도 1회 이상.

반복 제한 횟수보다 많은 값을 선언하면 무시됨.

“

font value example



Shorthand 'font' property

Name: **'font'**

Value: [[<'font-style'> || [<font-variant-css2>](#) || <'font-weight'> || [<font-stretch-css3>](#)]? <'font-size'> [/ [<line-height'>](#)]? <'font-family'>] | caption | icon | menu | message-box | small-caption | status-bar

Initial: see individual properties

Applies to: [all elements](#)

Inherited: yes

Shorthand 'font' value

(생략해도 괜찮은) 관심 없는 값 구문 제거:

```
[  
  [ <'font-style'> || <font-variant-css2> || <'font-weight'> || <font-stretch-css3> ]?  
  <'font-size'>  
  [ / <'line-height'> ]?  
  <'font-family'>  
] | caption | icon | menu | message-box | small-caption | status-bar
```

Shorthand 'font' value

(불필요한) 괄호 제거:

```
[  
  [ <'font-style'> || <'font-weight'> ]?  
  <'font-size'>  
  [ / <'line-height'> ]?  
  <'font-family'>  
]
```

Shorthand 'font' value

[<'font-style'> || <'font-weight'>]? <'font-size'> [/ <'line-height'>]? <'font-family'>

```
font: italic bold 16px/24px Sans-serif;  
font: bold italic 16px/24px Sans-serif;  
font: bold 16px Sans-serif;  
font: 16px Sans-serif;
```

⚠ 단축 속성에서 생략한 하위 속성 값들은 보통 초기값으로 설정된다.

“

지난 챕터 과제 해설



요구 사항

1. avif 이미지를 제공하세요. webp, jpeg 이미지를 대체 수단으로 제공하세요.
2. 960픽셀 미만 해상도에서 small 이미지를 출력하세요. 961픽셀 이상 해상도에서 large 이미지를 출력하세요.
3. HTML 코드만을 사용하고 1, 2 요구 사항을 동시에 대응하는 하나의 <picture> 요소를 구현하세요.

과제 소스

<https://naradesign.github.io/html5/img/small.avif>

<https://naradesign.github.io/html5/img/large.avif>

<https://naradesign.github.io/html5/img/small.webp>

<https://naradesign.github.io/html5/img/large.webp>

<https://naradesign.github.io/html5/img/small.jpg>

<https://naradesign.github.io/html5/img/large.jpg>

과제 해설

<https://t.ly/YZFf>

“

실습 과제



실습 과제

W3C 명세를 참고하여 CSS 퀴즈를 풀어 보세요.

<https://t.ly/8Rws>

퀴즈 해설은 다음 회차 강의 마지막에 공개합니다.

Thank you !

