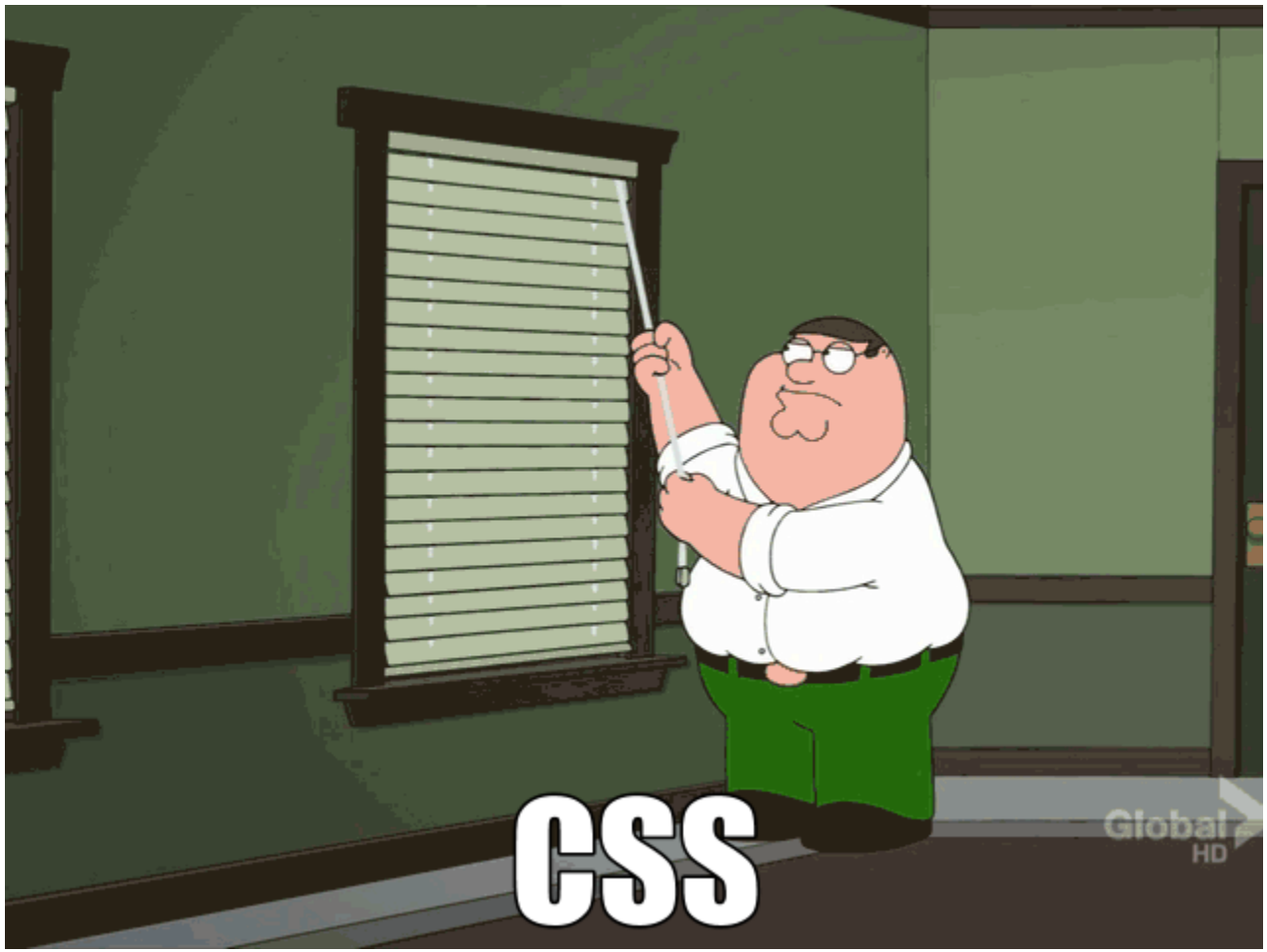
Quella sporca dozzina (a cascata)

Davide Di Pumpo — WebAppConf 2018 </>

Ma perchè?



in the name of the "separation of concerns",
ht minds and passionate developers

but it's limitations are staggering and the
esigners for many years to come unless

ogical failure. CSS just doesn't work as
s of sites use it? CSS can be used to style
t in how they use this technology. Even
artially adhere to the standard to truly be
lity and Consistency. CSS has neither. Any
plex site using CSS will tell you that all
g.

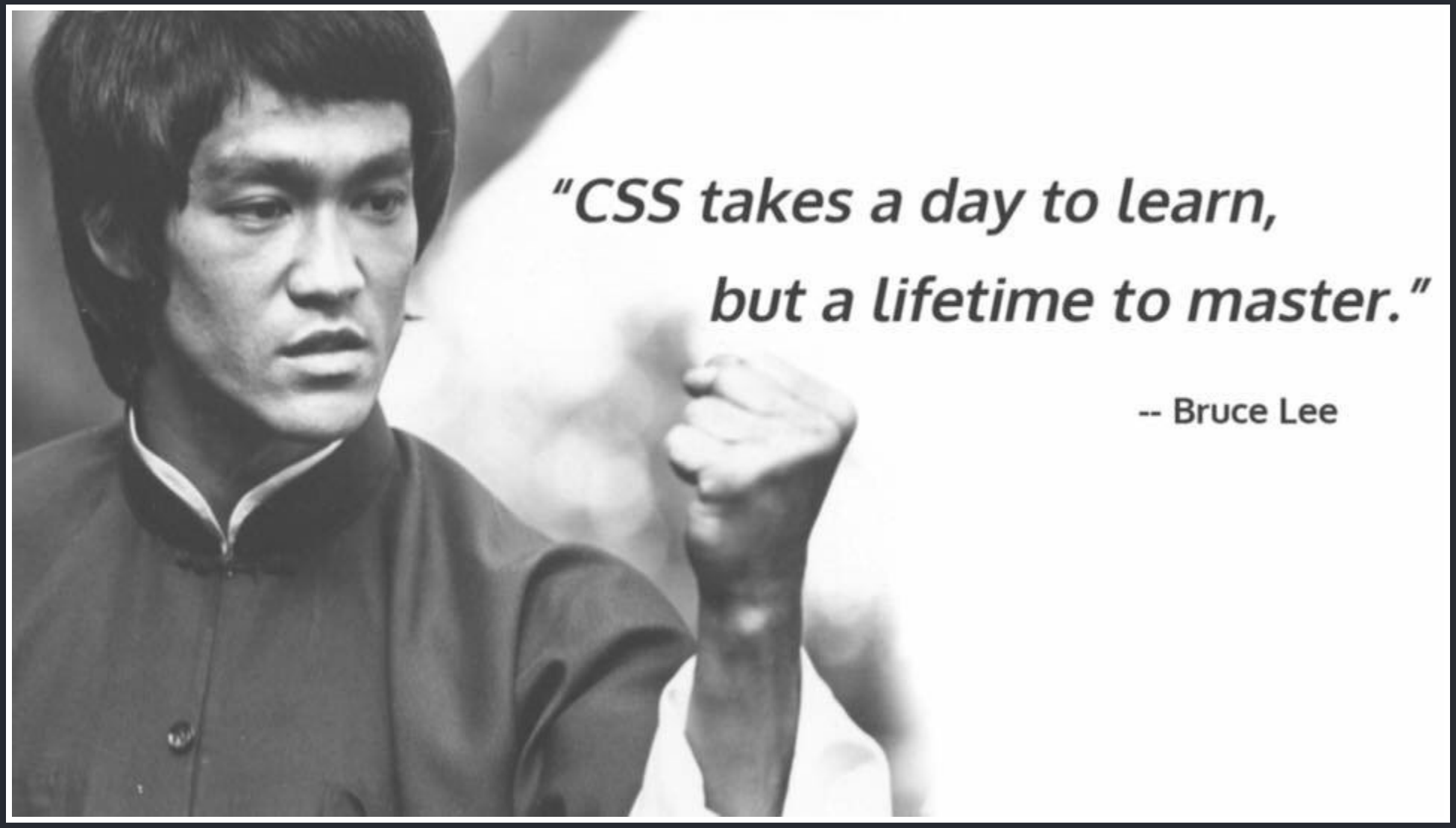
this idea that designers should spend
ime fiddling around with markup tags and
' I mean that every CSS design tool forces
tractive modern site. Many designers take
uld be design centric. PDF/postscript is a
unately not very suitable for the web.)
tically correct postscript tags they just
S sucks because it forces designers to
than how to make it work from a design

perspective.

BASTA!!1!

Make CSS great again!

Non stiamo parlando delle difficoltà del CSS



A tal riguardo...

Quindi?

E il CSS è divertente!

Sì ma sto titolo?

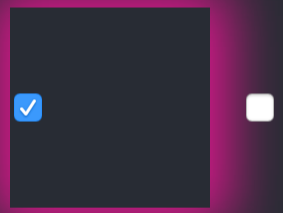
Let's start meow!!!

#1 :checked

```
input[type=checkbox]:checked {
    box-shadow: 0 0 50px 10px #E0168F;
}
```

```
<input type="checkbox" name="checkbox" id="checkbox1" checked="">
```

```
<input type="checkbox" name="checkbox2" id="checkbox2" checked="">
<input type="checkbox" name="checkbox3" id="checkbox3">
```



Rendiamolo interessante

Sibling combinator

```
sel ~ sel
```

```
input[type=checkbox]:checked ~ .selector {
  background: #E0168F;
}
```

Not checked

Demo

Checked

Demo

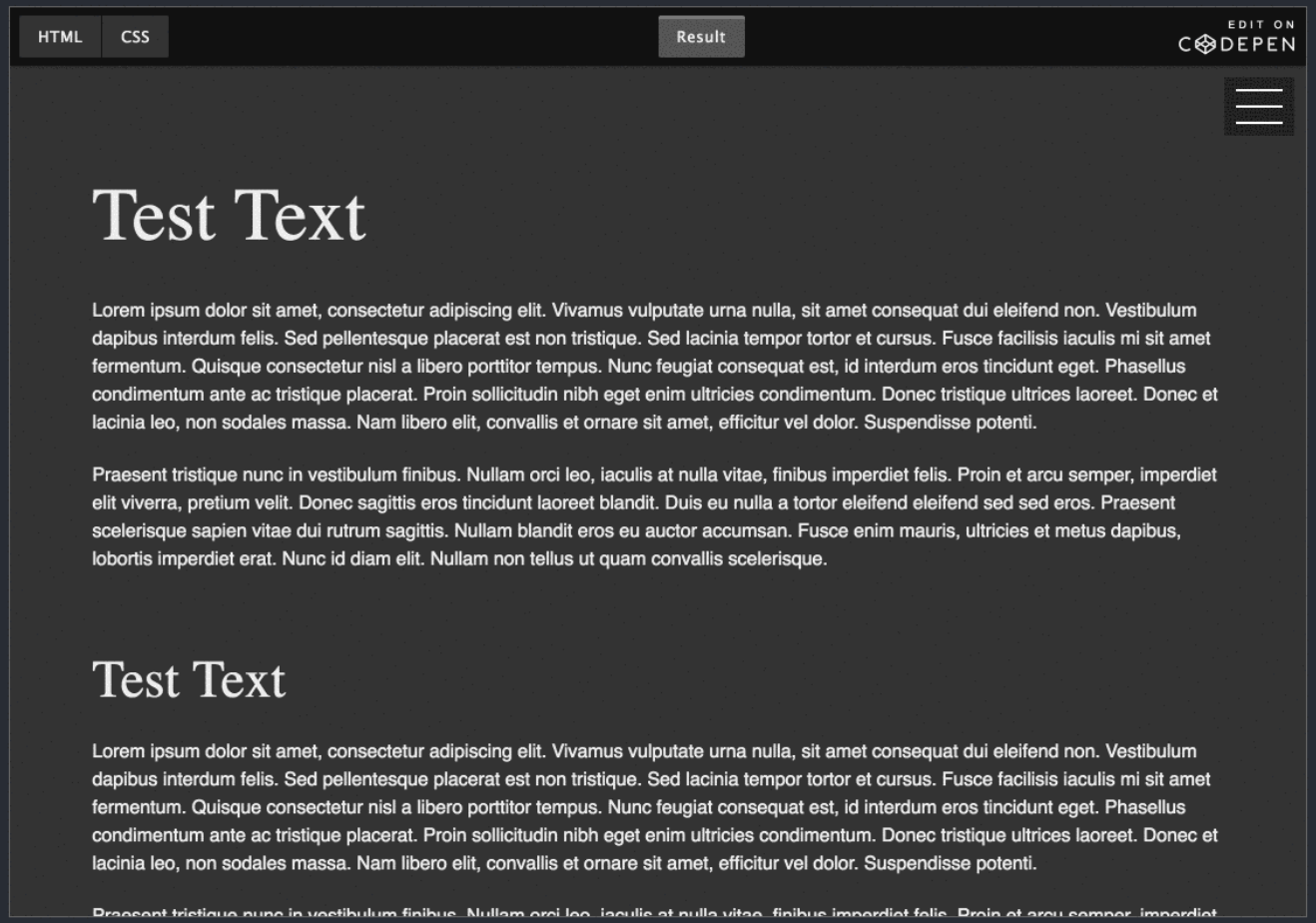
Ok, nella vita vera?

Custom checkbox!!!

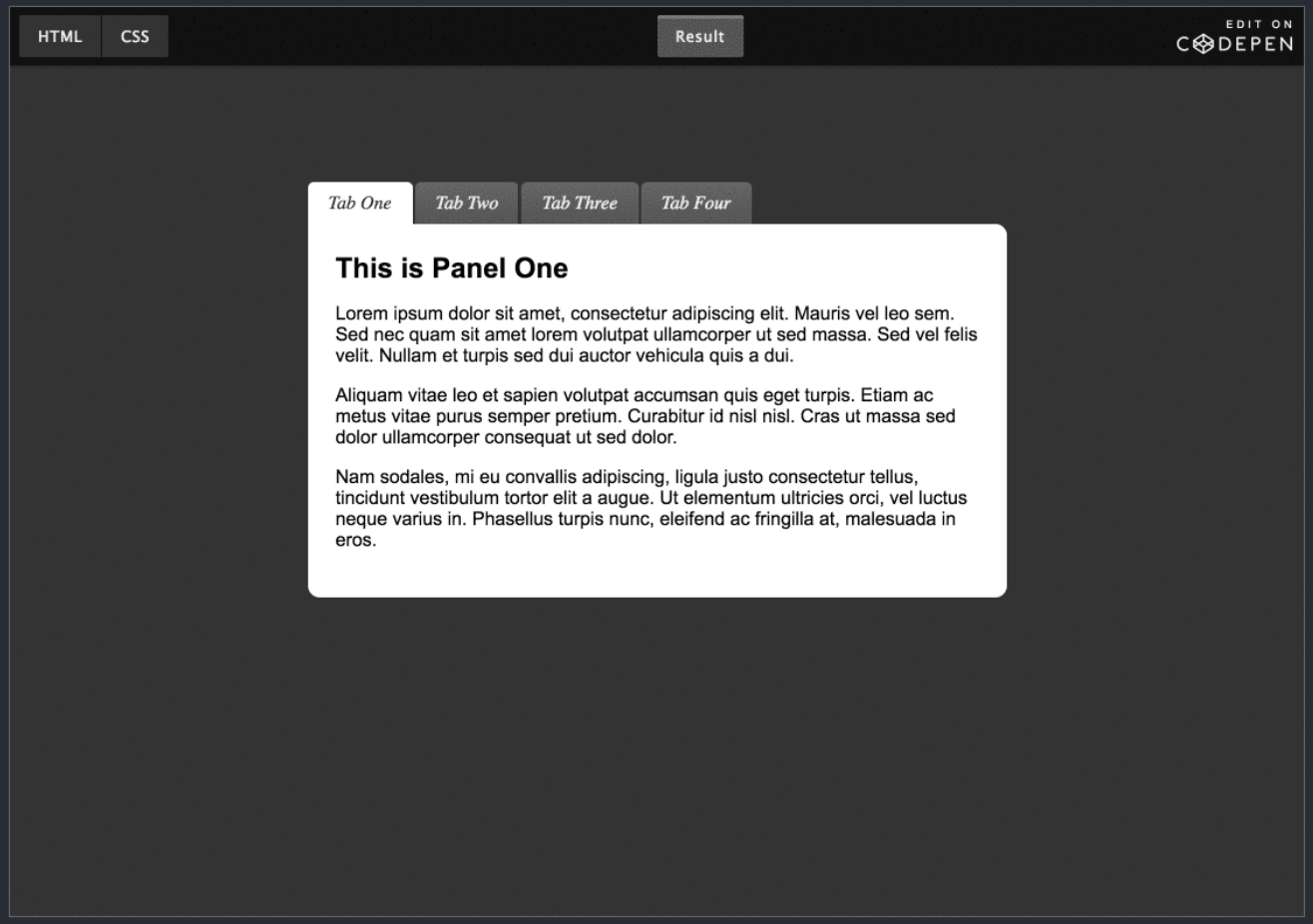
```
<div class="cb">
  <div class="cb__cont">
    <input id="cbId1" type="checkbox" class="cb__input" checked="">
    <svg version="1.1" stroke-miterlimit="10" xmlns="http://www.w3.or
      <path fill="transparent" stroke="#000000" stroke-width="4" stro
    </svg>
    <div class="cb__background"></div>
  </div>
  <label for="cbId1" class="cb__label">
    What a wonderful checkbox
  </label>
</div>
```

What a wonderful checkbox

Ma praticamente potete gestire qualsiasi stato binario



See the Pen Pure CSS3 Off Canvas Menu by Terryl (@Terryl_Brown) on CodePen.



See the Pen CSS Only Tabbed Content by Stephen Greig (@stephengreig) on CodePen.

HTML CSS Result EDIT ON CODEPEN

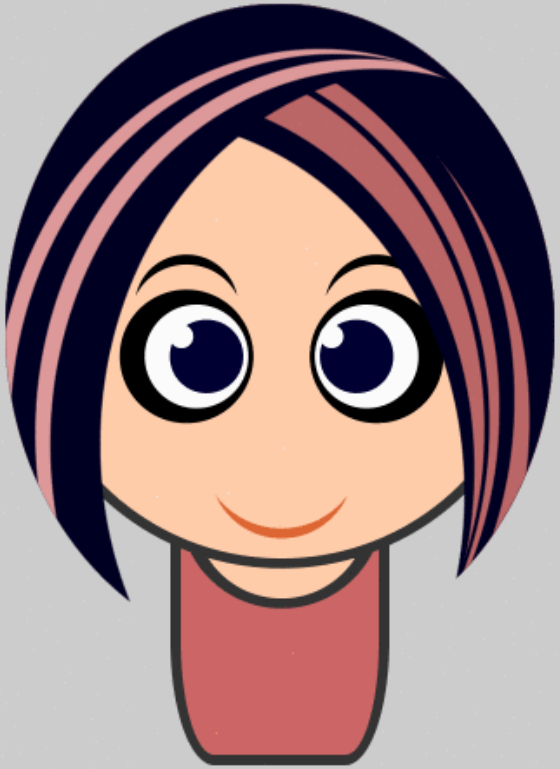
One element girl

expression

Normal Smile Angry

color

Pink Blue Green



See the Pen One Element Girl by keisuke Takahashi (@ksksoft) on CodePen.

Accessibilità?

- Il toggle deve avere un attributo `aria-expanded` con lo stato corrente
- Il toggle deve avere un attributo `aria-controls` che indichi cosa controlla
- Nel caso di contenuto nascosto, l'elemento deve avere un attributo `aria-hidden` che ne indichi lo stato

Cattiva notizia:

Ci vuole javascript per farlo 😣

Buona notizia:

C'è un js minuscolo (600 Byte) che lo fa in automatico 😃

a11y-toggle

#2 :empty

```
div:empty {
  border: red;
}
```

```
<div></div> <!-- match -->

<div><!-- match --></div>

<div> </div><!-- nope -->

<div>
</div><!-- nope -->
```

Bello ma che ci faccio?

- Nascondere elementi vuoti
-
- Tipo quello qua sopra

```
li:empty {display: none}
```

- Nascondere elementi vuoti
- Tipo quello qua sopra

Segnalare errori

```
<div class="main"><!--Your generated by JS app will go here--></div>
```

```
.main:empty {
  background: #e74c3c;
  padding: 1em;
  margin: 1em 0;
  border-left: 5px solid #c0392b;
}

.main:empty::before {
  content: "The app is unavailable now!"
}
```

The app is unavaible now!

Utilizzarlo per gestire il loading

```
<div class="main2"><!--Your generated by JS app will go here--></div>
```

```
.main2:empty::after {
  display: block;
  content: '';
  color: #ffffff;
  font-size: 90px;
  text-indent: -9999em;
  overflow: hidden;
  width: 1em;
  height: 1em;
  border-radius: 50%;
  margin: 72px auto;
  position: relative;
  transform: translateZ(0);
  animation: load6 1.7s infinite ease, round 1.7s infinite ease;
}
```

O perchè no, fare entrambe le cose contemporaneamente!

In combo con :not

Load content

```
.main4:not(:empty) {
  animation: empty-fade 1s, empty-translate-in 1s;
}
```

#3 :target

`:target` è una CSS pseudo-class che seleziona un elemento univoco (l'elemento target) che abbia l'id che corrisponde a quello nell'url.

Già le vecchie ancore!

```
<a href="#pippo">Pippo Link</a>
<div id="pippo">PIPPO!!1!</div>
```

```
div:target {
  background: #E0168F;
  color: #fff;
}
```

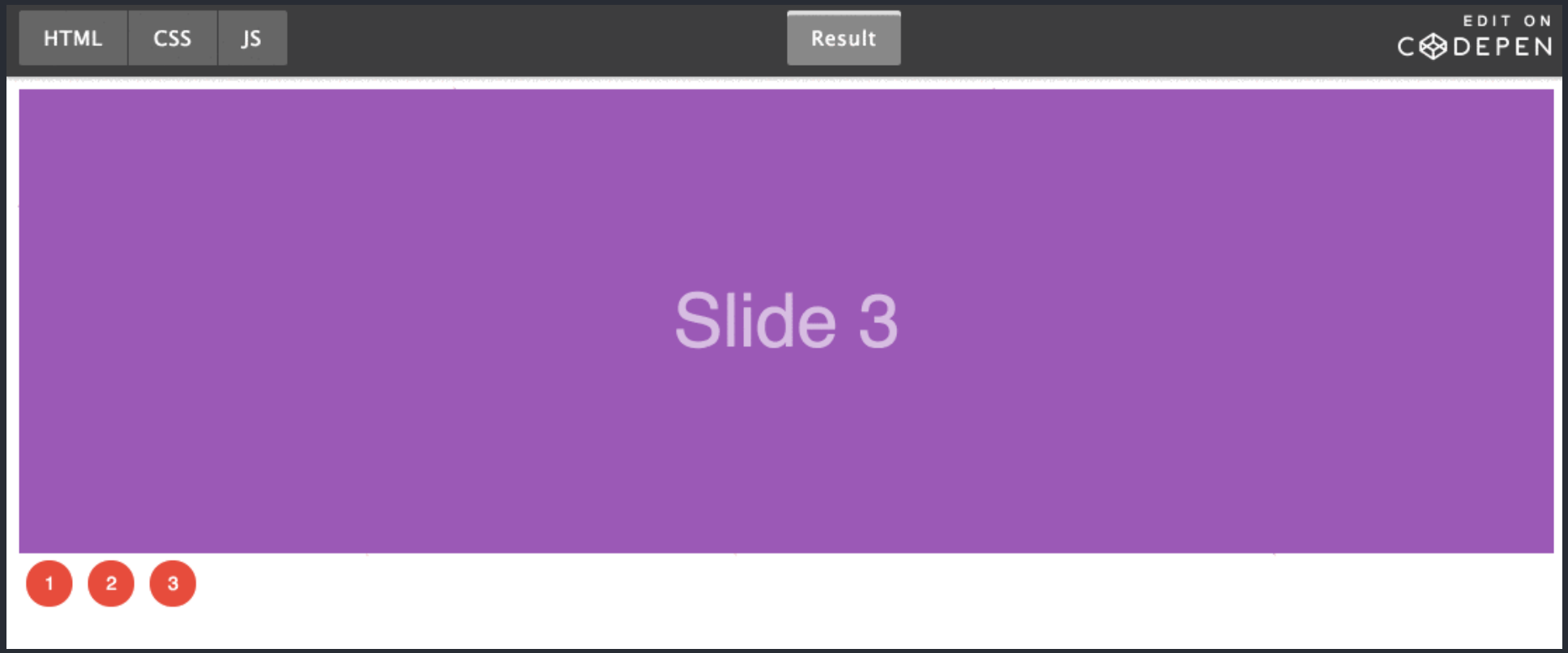
See the Pen target example by Davide Di Pumpo (@MakhBeth) on CodePen.

Eh ma come la mettiamo con il jump to?

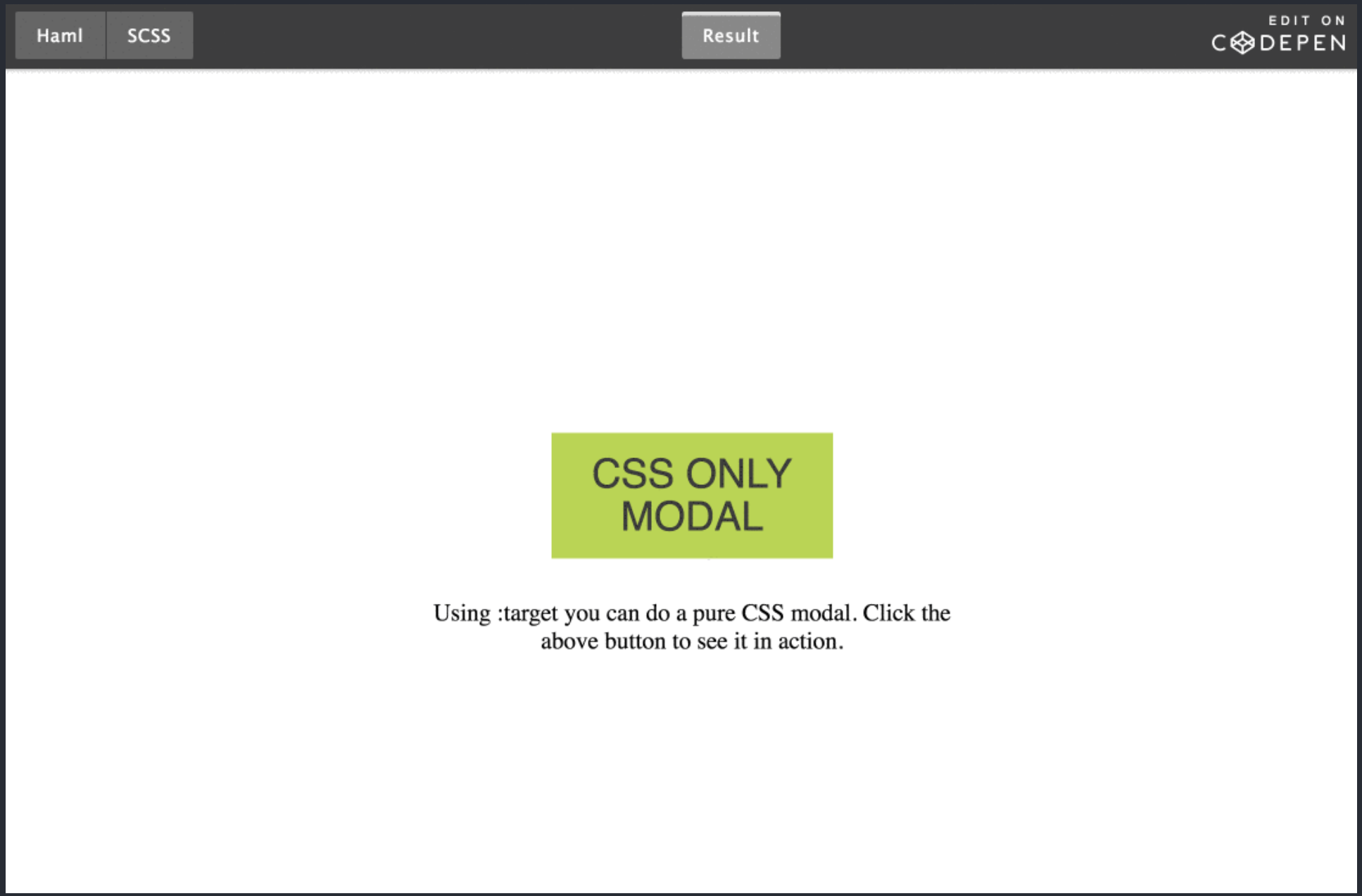
```
<a href="#pippo">Pippo Link</a>
<div class="target" id="pippo"></div>
<div>PIPPO!!1!</div>
```

```
.target {
  position: fixed;
  top: 0; left: 0;}
div:target + div {
  background: #E0168F;
  color: #fff;}
```

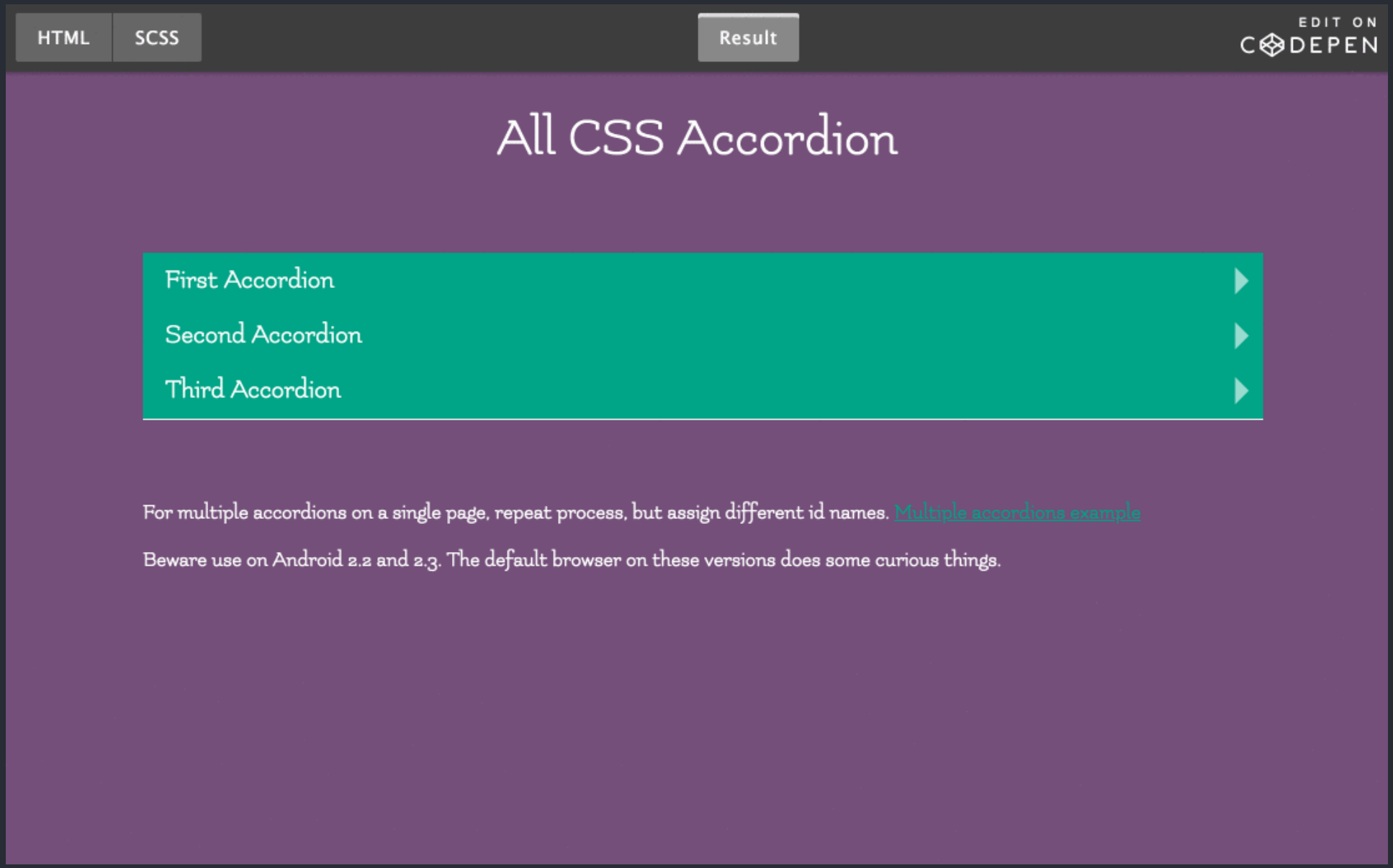
Via di esempi reali!!1!



See the Pen Simple Slider using :target by The Curious Developer (@thecuriousdev) on CodePen.



See the Pen CSS only modal by Jeff Ayer (@DeptofJeffAyer) on CodePen.



See the Pen CSS Accordion by pollardld (@pollardld) on CodePen.

#4 :nth

Ci sono un sacco di trucchi che potete fare con:

- `:nth-child`
- `:nth-of-type`
- `:nth-last-child`
- E le combinazioni varie con ~ e +

```
ul li:nth-last-child(n+3), ul li:nth-last-child(n+3) ~ li { }
```

Try it out

Your quantity query will be reflected on the items below by a change in colour. Add and remove items to see the styling be applied when the query matches.

ADD ITEM REMOVE ITEM

Un altro trucco che adoro è:

1. Item
2. Item
3. Item
4. Item
5. Item
6. Item
7. Item

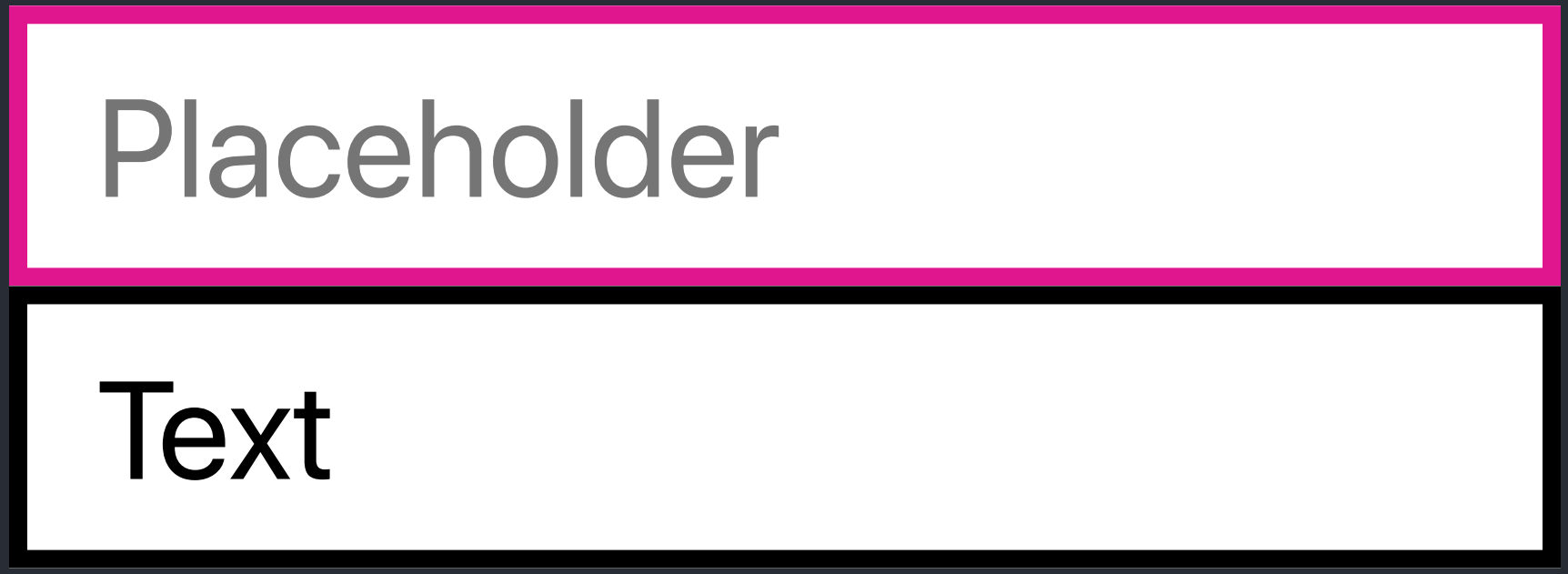
```
ol li:nth-child(-n+5):nth-child(n+3) {
  background: #E0168F;
}
```

#5 :placeholder-shown

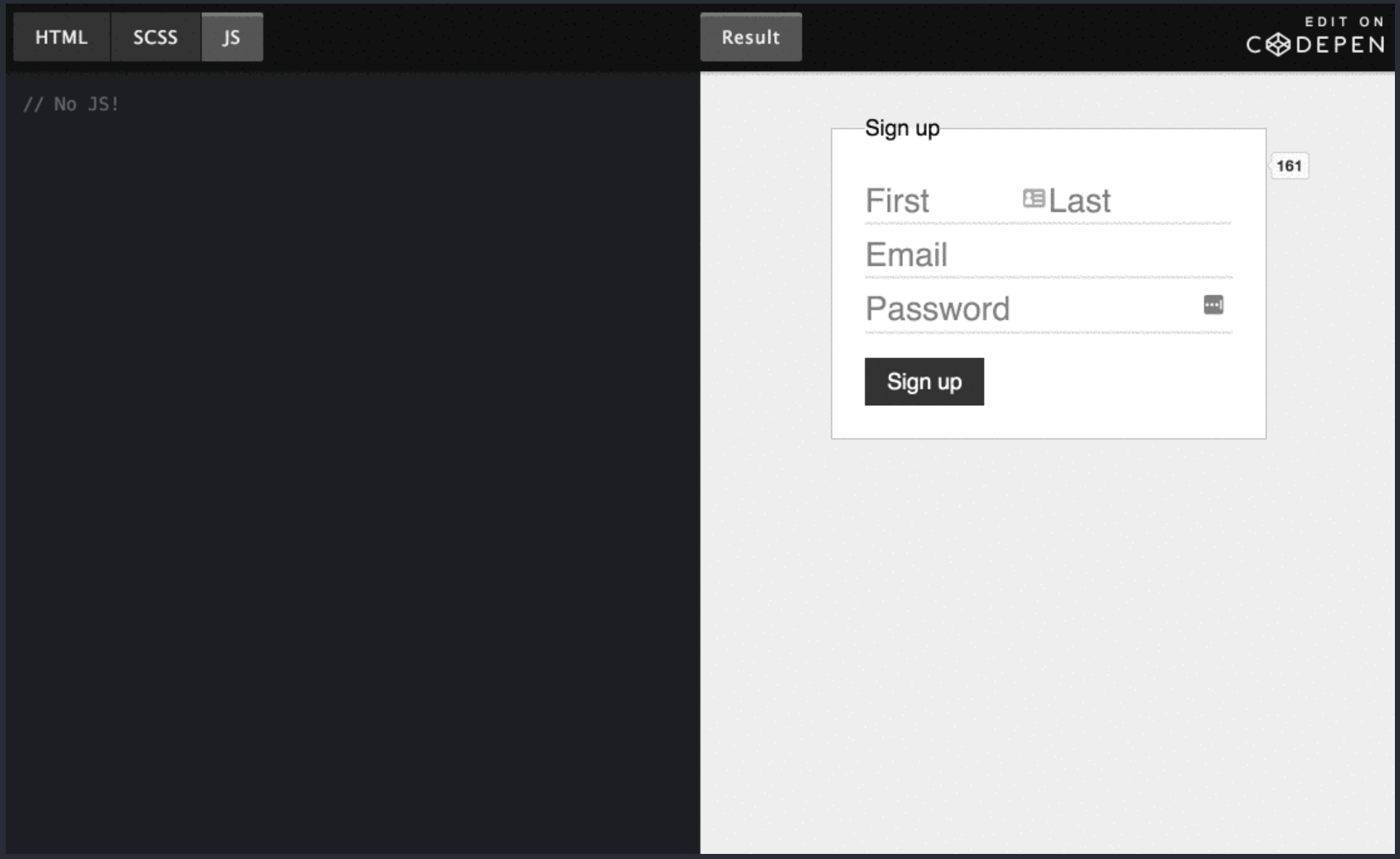
Un semplice selettore che vi dice se il placeholder di un input è visibile o meno!

```
<input type="text" placeholder="Placeholder">
<input type="text" placeholder="Placeholder" va
```

```
input {
  border: 2px solid black;
}

input:placeholder-shown {
  border-color: #E0168F
}
```


Placeholder
Text

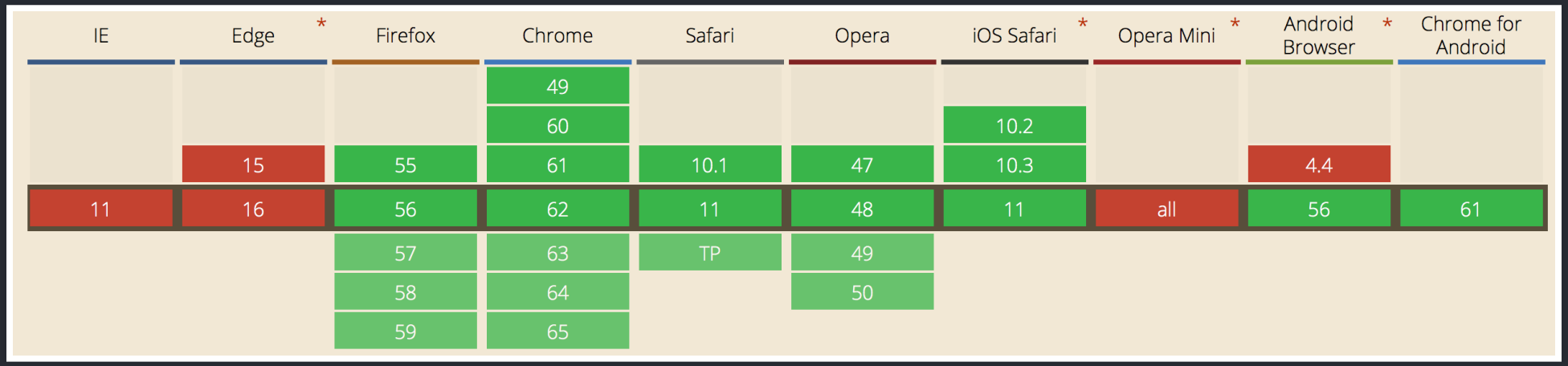
Un utilizzo reale?

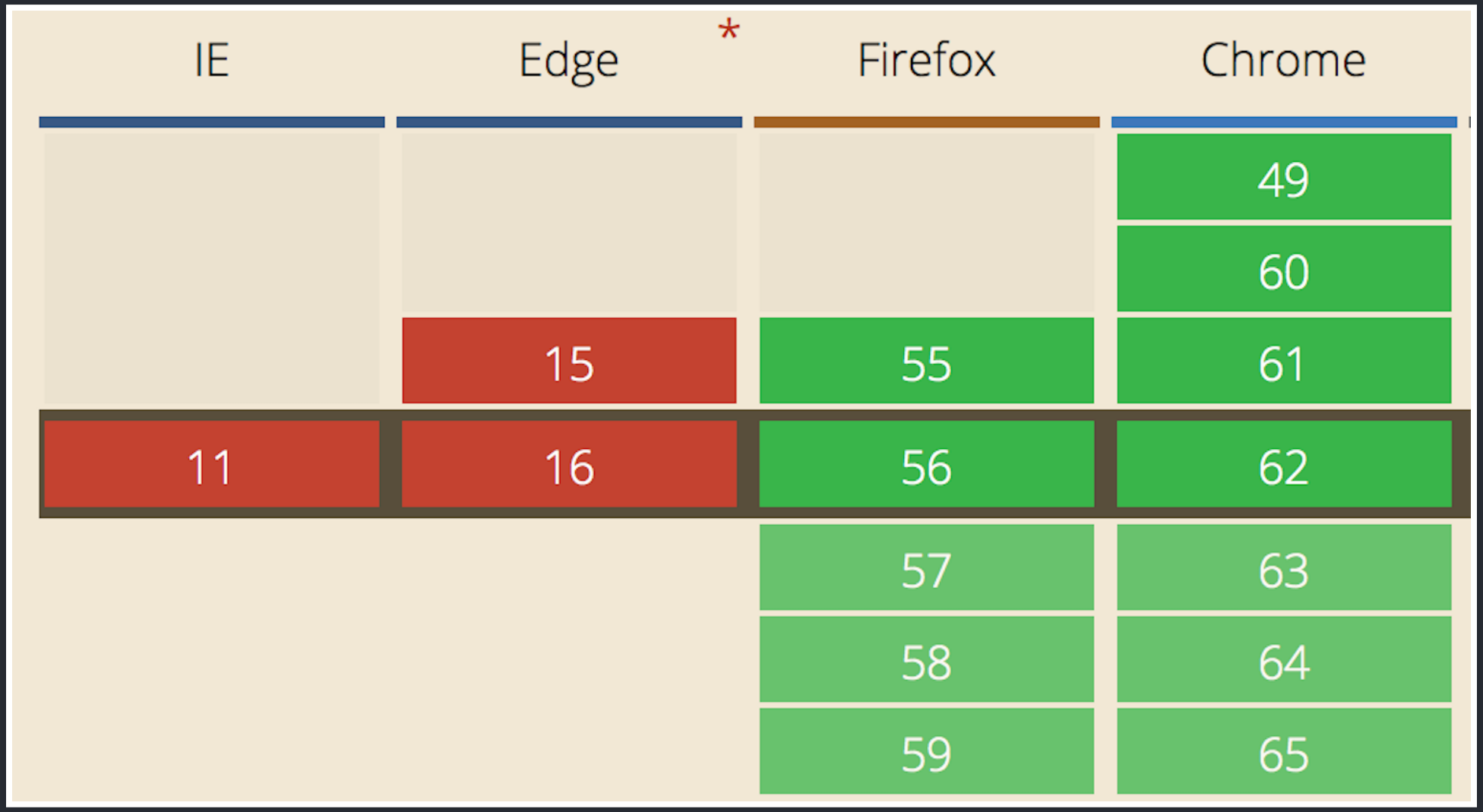


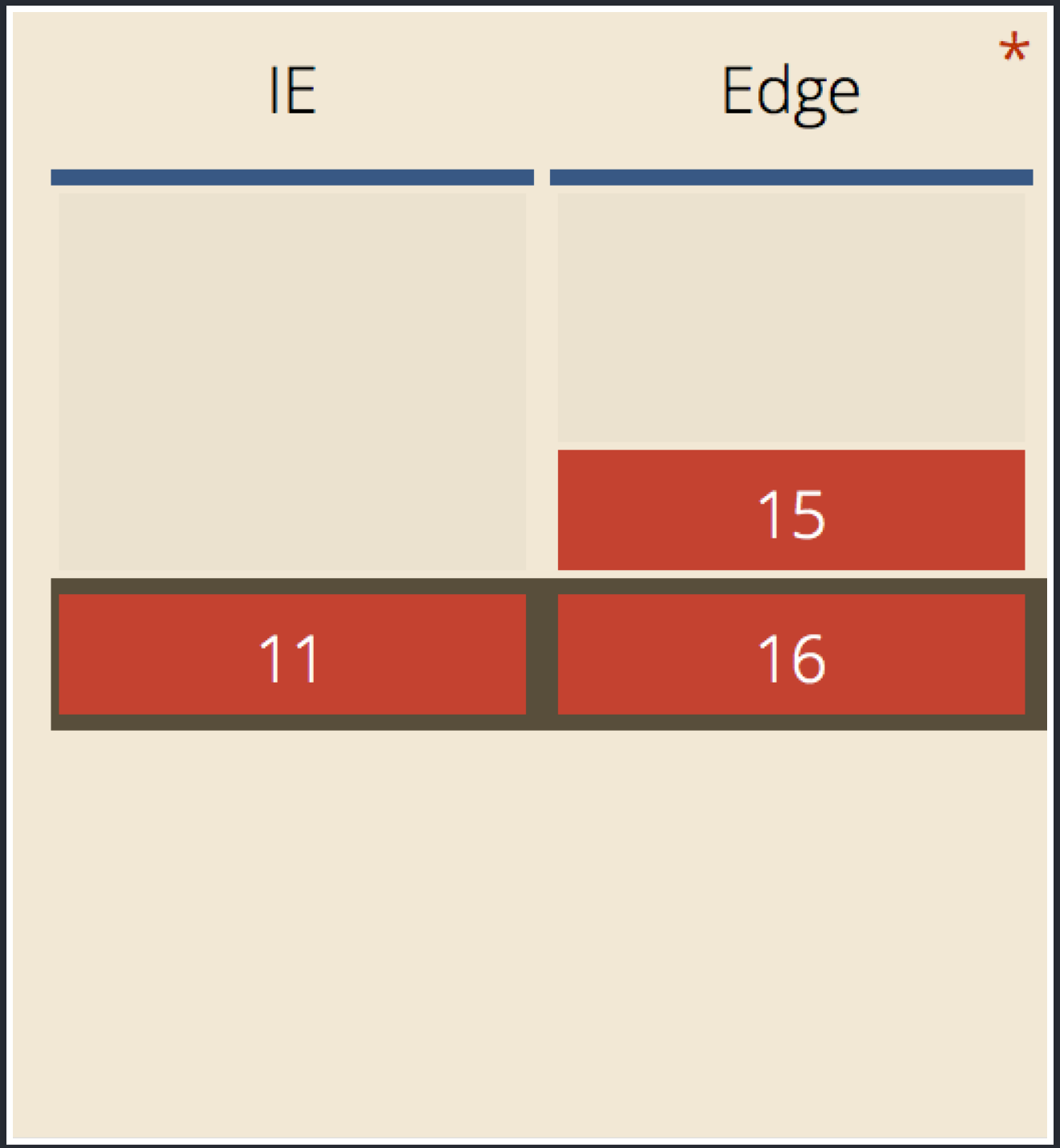
See the Pen Pure-CSS Float Label. Finally. by Anton Staroverov (@tonystar) on CodePen.

Semplice facile e veloce, no?

ma...


IE
Edge *
Firefox
Chrome
Safari
Opera
iOS Safari *
Opera Mini *
Android Browser *
Chrome for Android
49
60
10.2
15
55
61
10.1
47
10.3
4.4
11
16
56
62
11
48
11
all
56
61
57
63
TP
49
58
64
50
59
65


IE
Edge *
Firefox
Chrome
49
60
15
55
61
11
16
56
62
57
63
58
64
59
65


IE
Edge
*
15
11
16

MA...

... esiste un polyfill così piccolo da stare in una slide!

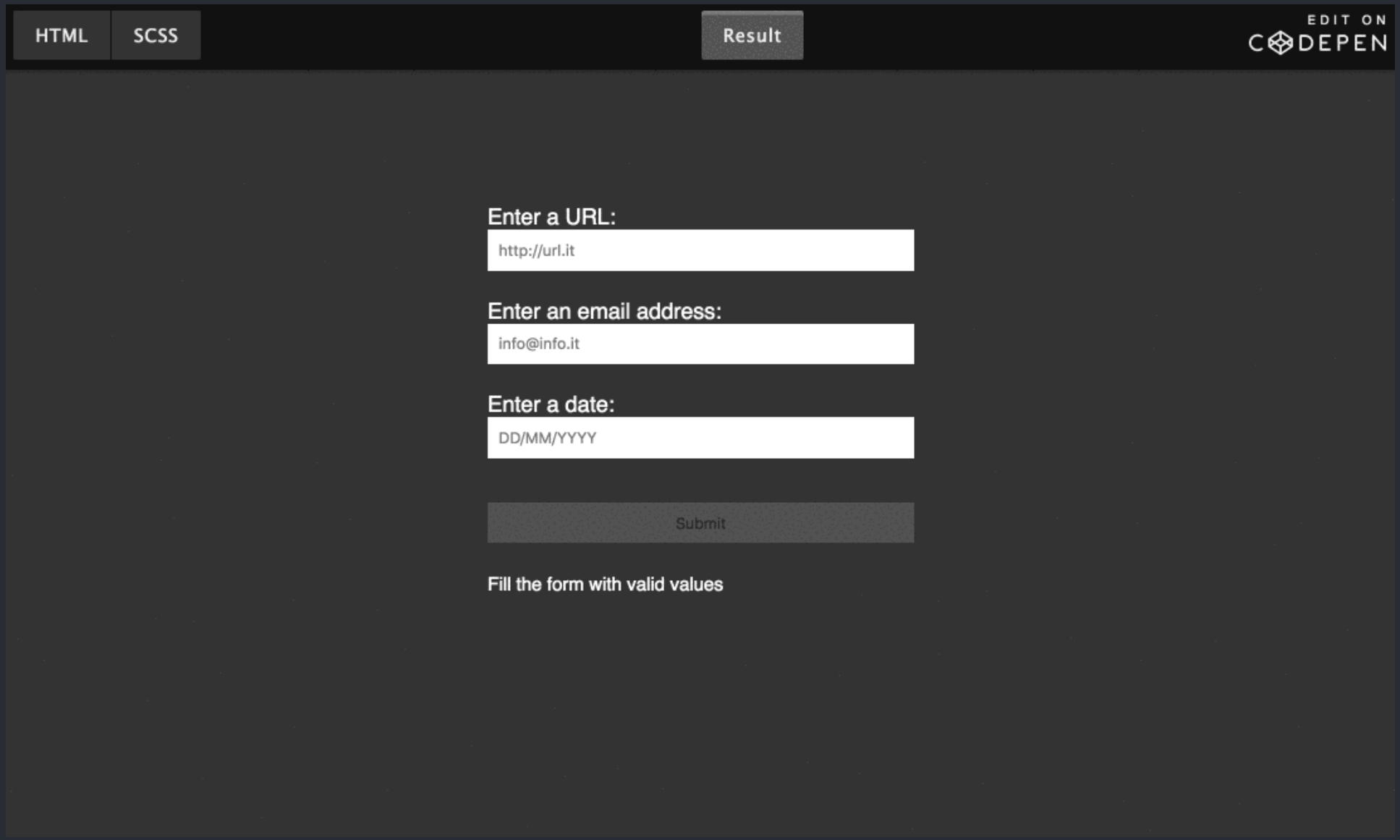
```
function placeholderPolyfill() {
  this.classList[this.value ? 'remove' : 'add']('placeholder-shown');
}

document.querySelectorAll('[placeholder]').forEach(el => {
  el.addEventListener('change', placeholderPolyfill);
  el.addEventListener('keyup', placeholderPolyfill);
});
```

#6 :invalid

Seleziona un input o un form non validi (tramite gli attributi HTML5)

In combo con gli altri selettori di form si possono stilare gli stati di un form

See the Pen Example Form 1 by Davide Di Pumpo (@MakhBeth) on CodePen.



See the Pen Example Form 1 by Davide Di Pumpo (@MakhBeth) on CodePen.

Cose negative?

No serio, io non le ho trovate 😁

Siamo solo a metà

Acceleriamo!

#7 currentColor

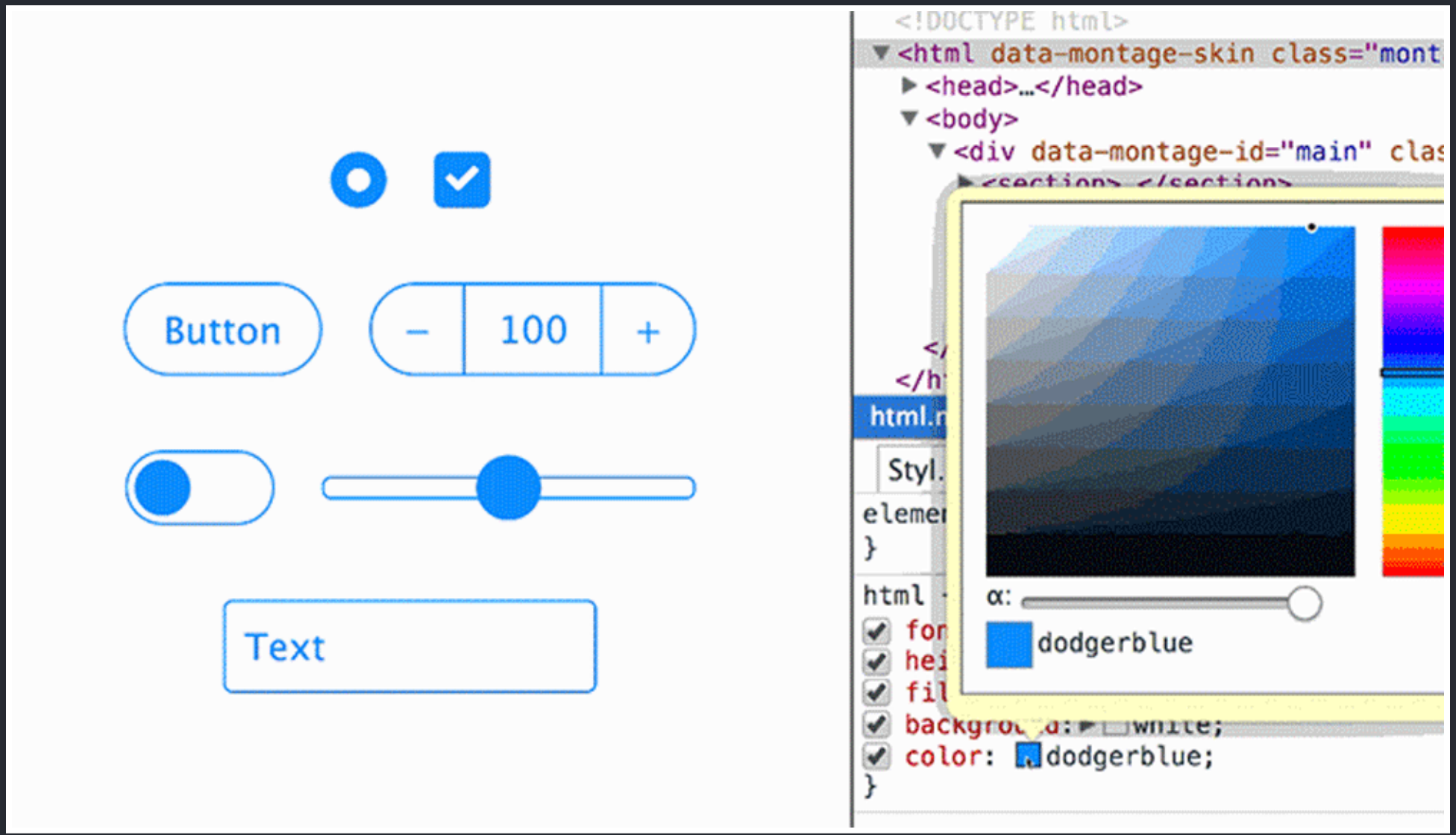
È una variabile sempre valorizzata con il valore della propietà `color` corrente

```
<div class="example-currentColor">Example</div>
<style>
  .example-currentColor{
    color: #E0168F;
    border-bottom: 0.5em solid currentColor;
    box-shadow: 0px 0px 1px currentColor;
  }
</style>
```

Example

Real life?


```
.example-currentColor-1 button {
  background: transparent;
  border: 2px solid currentColor;
}
.example-currentColor-1 button.info {
  color: #3498db;
}
.example-currentColor-1 button.success {
  color: #27ae60;
}
.example-currentColor-1 button.error {
  color: #c0392b;
}
```


Button
100
Text
<!DOCTYPE html>
<html data-montage-skin class="mont
<head>…</head>
<body>
<div data-montage-id="main" clas
α:
dodgerblue
color: dodgerblue;
}

Il mio preferito è con le SVG!

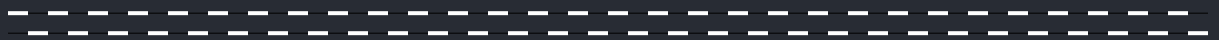

```
.example-currentColor-1 svg {
  fill: currentColor;
}
```

#8 SVG path

Ci sono due proprietà `stroke-dasharray` e `stroke-dashoffset` che si riferiscono allo stroke di una svg

`stroke-dasharray` indica come tratteggiare la linea (un valore di 5 creerà dei `dash` di 5px distanziati da loro di 5px

`stroke-dashoffset` indica da dove, dall'inizio dell'svg si inizia a tratteggiare.



```
<style>
  line.example-svg-1,
  line.example-svg-2 {
      stroke: white;
      stroke-width: 2;
      stroke-dasharray: 10;
  }
  line.example-svg-2 {
      stroke-dashoffset: 10;
  }
</style>
<svg width="600" height="40" viewBox="0 0 600 40" version="1.1" xmlns
  <line class="example-svg-1" x1="0" y1="10" x2="600" y2="10"></line>
  <line class="example-svg-2" x1="0" y1="20" x2="600" y2="20"></line>
</svg>
```

```
@keyframes example-svg-3 {
  0% {
    stroke-dashoffset: 600;
  }
  100% {
    stroke-dashoffset: 0;
  }
}
```

nella vita reale?

HTML SCSS

```scss
.path-j {
  stroke-dasharray: 1500;
  stroke-dashoffset: 1500;
  animation: dash 0.8s ease-out forwards;
}

.path-am {
  stroke-dasharray: 1000;
  stroke-dashoffset: 1000;
  animation: dash 0.9s ease-out forwards 0.86s;
}

.path-e {
  stroke-dasharray: 1000;
  stroke-dashoffset: 1000;
  animation: dash 0.6s ease-in forwards 1.5s;
}

.path-s {
  stroke-dasharray: 1000;
  stroke-dashoffset: 1000;
  animation: dash 0.69s ease forwards 2.3s;
```

Result

EDIT ON CODEPEN

See the Pen What's My Name? by James Nowland (@jnowland) on CodePen.

Ma il mio esempio preferito?

 What a wonderful checkbox

 Broken image

#9 Broken images

> *Potete fare le UI più fighe dell'universo, ma quando si spacca un'immagine ve voglio vedè a fa i ganzi*

Confucio

```css
img { min-height: 50px; position: relative;
  display: inline-block; text-align: center;}

img::before, img::after {
  display: block; position: absolute;
  top: 50%; left: 50%;
  transform: translate(-50%, -50%);}

img:before {
  content: " ";
  height: 100%; width: 100%;
  background-color: #ecf0f1;
  border: 10px dashed currentColor;}

img:after {
  content: "😭 Image not available: " attr(alt);
  color: #7f8c8d;}
```

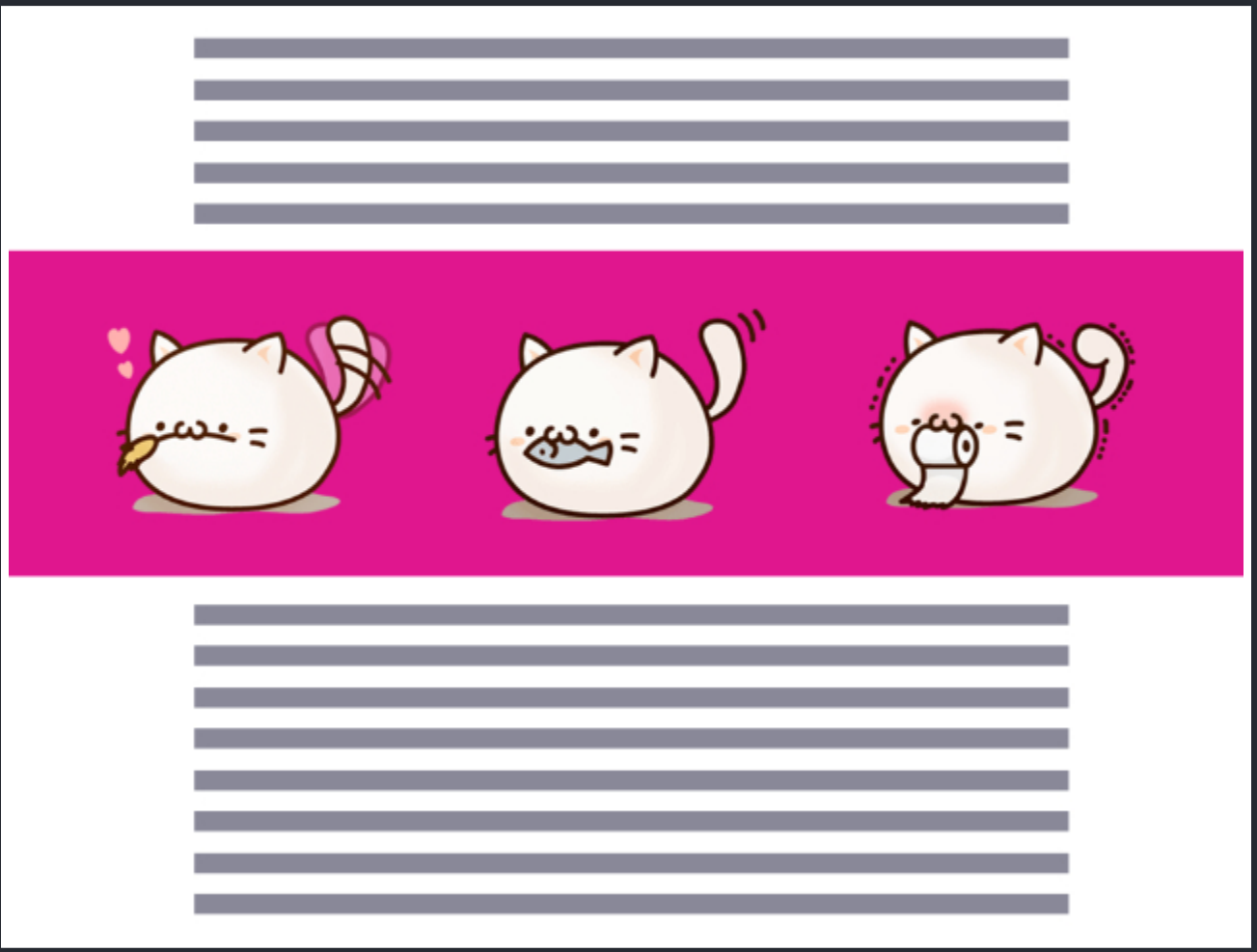

😭 Image not available:
Awesome stuff

Due cose:

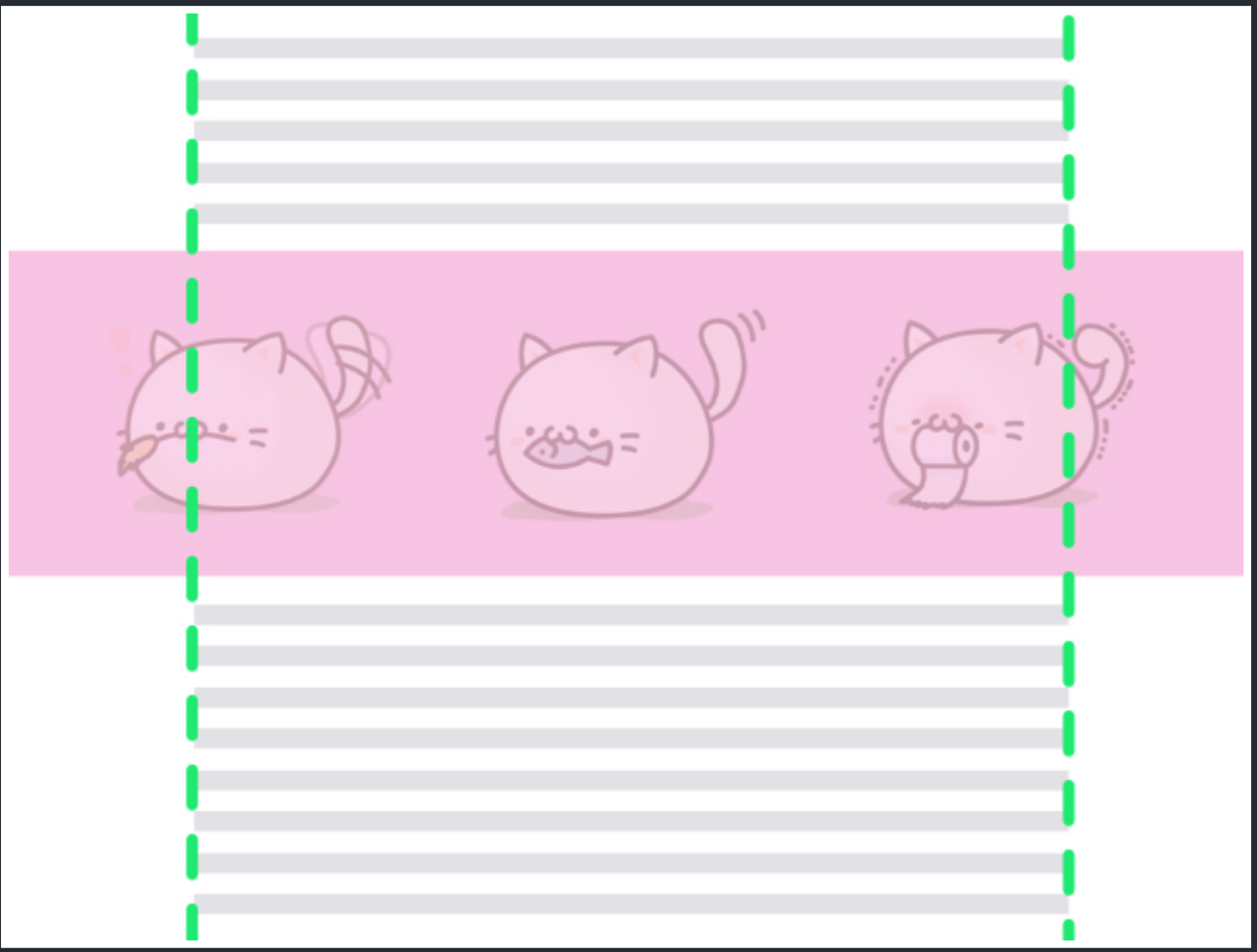
- `content: "..." attr(alt);`
- Le brutte notizie

Sto trucco non funziona su IE e Safari...

#10 Escape the cage



Cats are frome here



Lorem Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi id semper quam. Ut sodales venenatis enim, eget vestibulum purus commodo non. Etiam convallis mi sapien, eu mollis velit aliquam ac. Sed fringilla quis ipsum eget pharetra. Nullam commodo ligula sed mattis ornare. Sed ullamcorper luctus diam vel vehicula. Nulla facilisi. Phasellus orci est, tempor et diam a, consectetur interdum metus. Aliquam erat volutpat. Praesent porttitor velit eu fringilla condimentum. Curabitur sed massa dignissim elit vulputate tincidunt in ac eros. Maecenas a elementum leo, ut elementum risus. Morbi id pulvinar turpis. Duis posuere sapien libero, et tempus neque consequat sit amet.

Sed lacus dolor, blandit et tempor at, consectetur luctus velit. Praesent a dapibus nunc. Duis vel orci vel justo sollicitudin accumsan. Proin convallis sapien ac arcu egestas vehicula. Maecenas nulla elit, volutpat eu nibh at, dictum fringilla dui. Aenean pharetra nisi feugiat rhoncus fringilla. Ut varius semper pharetra. Aenean non finibus ligula. Pellentesque consectetur ac quam ac sodales.

```
<div class="example-cage">
  <div class="e-container">
    Lorem Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mo

    <div class="our-div"></div>

    Sed lacus dolor, blandit et tempor at, consectetur luctus velit.
  </div>
</div>
```

Lorem Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi id semper quam. Ut sodales venenatis enim, eget vestibulum purus commodo non. Etiam convallis mi sapien, eu mollis velit aliquam ac. Sed fringilla quis ipsum eget pharetra. Nullam commodo ligula sed mattis ornare. Sed ullamcorper luctus diam vel vehicula. Nulla facilisi. Phasellus orci est, tempor et diam a, consectetur interdum metus. Aliquam erat volutpat. Praesent porttitor velit eu fringilla condimentum. Curabitur sed massa dignissim elit vulputate tincidunt in ac eros. Maecenas a elementum leo, ut elementum risus. Morbi id pulvinar turpis. Duis posuere sapien libero, et tempus neque consequat sit amet.

Sed lacus dolor, blandit et tempor at, consectetur luctus velit. Praesent a dapibus nunc. Duis vel orci vel justo sollicitudin accumsan. Proin convallis sapien ac arcu egestas vehicula. Maecenas nulla elit, volutpat eu nibh at, dictum fringilla dui. Aenean pharetra nisi feugiat rhoncus fringilla. Ut varius semper pharetra. Aenean non finibus ligula. Pellentesque consectetur ac quam ac sodales.

```css
.our-div.free {
  width: 100vw;
  position: relative;
  left: 50%;
  transform: translateX(-50%);
}
```

Ci sono altre soluzioni che prevedono `calc` che potete trovare online, simili a:

```
.our-div.free {
  margin: 0 calc(-50vh + 50%);
}
body, html {
  overflow: hidden;
}
```

Non le adoro a causa di quell'overflow hidden;

#11 Lavorare con la legacy

Cambiate lavoro e:

```
<div class="col cols red big col-m-2 flex fortune clearfix"></div>
```

Combattere il male con il male

Potete raggruppare i selettori con caratteri non validi:

```
<div class="col-sm-6 col-md-2 red big"></div>
```

```
<div class="
  [ #bootstrap col-sm-6 col-md-2 ]
  [ #utility red big ]
  my-selector
">
</div>
```

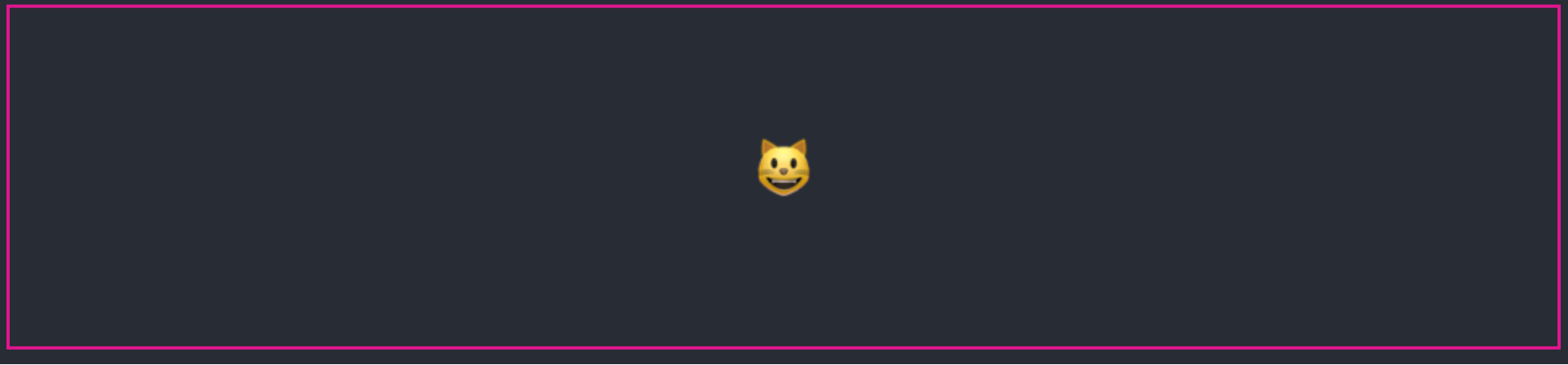
```
./#utility {
  border: 1px solid red !important;
}
```

Avete paura del global scope?

Usate le emoji...

Le emoji sono selettori CSS validi

```
<style>.😺 {
border: 2px solid #E0168F;
color: white;
padding: 5rem;
text-align: center;
}</style>
<div class="😺">😺</div>
```



E ricordate che prima del `!important`

Potete moltiplicare le classi per aumentare la specificità

```
<div class="example-double">Colore!</div>
<style>
  .example-double.example-double {
    background-color: #E0168F;
  }

  .example-double {
    background-color: red;
  }
</style>
```

Colore!

#12 Bonus track

Potete usare `::before` e `::after` con elementi autochiudenti che non effettuano replace

---

See the Pen YEqavq by Davide Di Pumpo (@MakhBeth) on CodePen.

Grazie ad un selettore come `a:not([href*="mio-sito.cool"])` potete segnalare tutti i link esterni al vostro sito

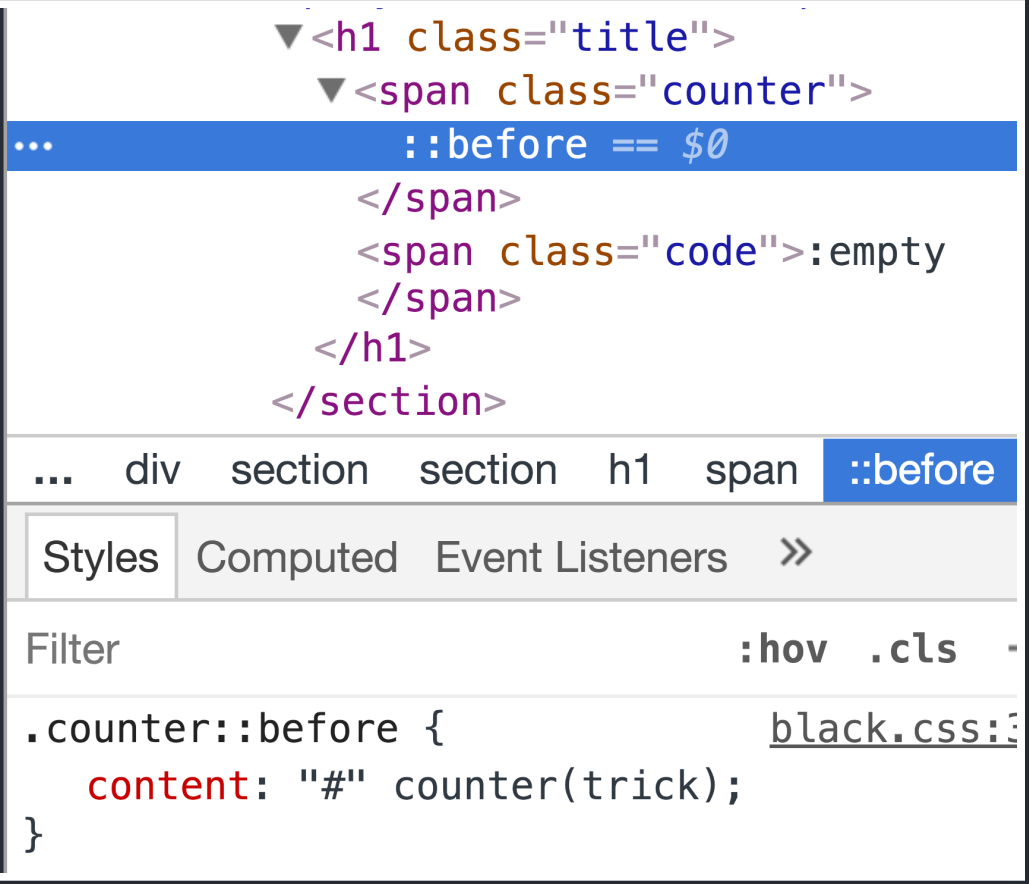
Link interno
Link a bellissimi siti di gatti

```
a:not([href*="localhost"]) {
  color: red;
}
```

Con le proprietà CSS `counters` potete numerare oggetti con il solo CSS

Un esempio reale?

#1 :empty


<h1 class="title">

::before == $0

:empty

</h1>
</section>
... div section section h1 span ::before
Styles Computed Event Listeners »
Filter :hov .cls
.counter::before { black.css:
content: "#" counter(trick);
}

Davide Di Pumpo

- ~~Web Designer~~
- Senior Front-End Developer Credimi
- Co-organizzatore Milano Frontend
- **MakhBeth** su: Twitter, Github, Internet


THANK YOU