

Quella sporca dozzina

(a cascata)

Davide Di Pumbo – {codemotion}

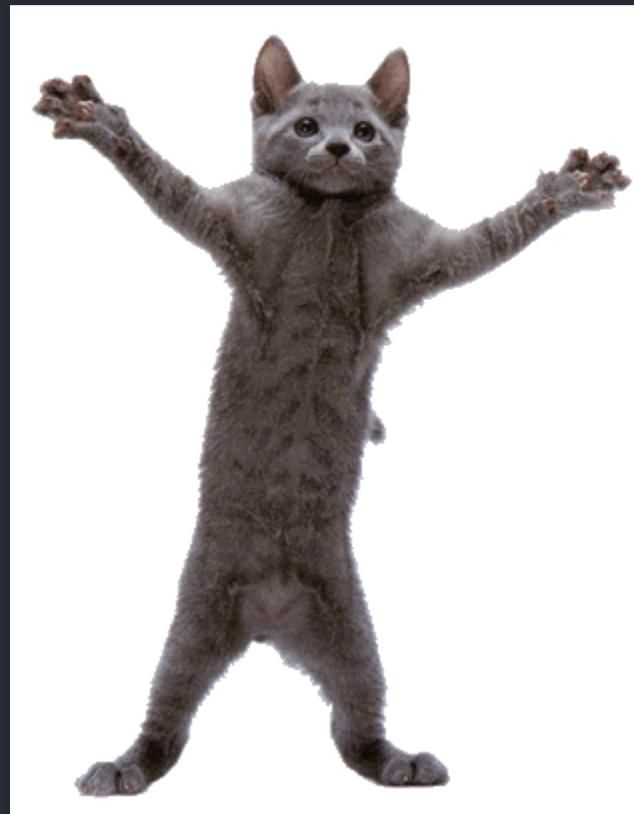
Chi sono?



Davide Di Pumpo

- ~~Web Designer~~
- Senior Front-End Developer [Credimi](#)
- Co-organizzatore [Milano Frontend](#)
- **MakhBeth** su: [Twitter](#), [Github](#), [Internet](#)
- Amo Gatti, Malti, Fumetti e Giochi...

Let's start meow!!!



Ma perchè?



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



but its limitations are staggering and the
designers for many years to come unless

logical failure. CSS just doesn't work as
sites use it? CSS can be used to style
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
j-

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
uncately not very suitable for the web.)
tically correct postscript tags they just
sucks because it forces designers to
than how to make it work from a design

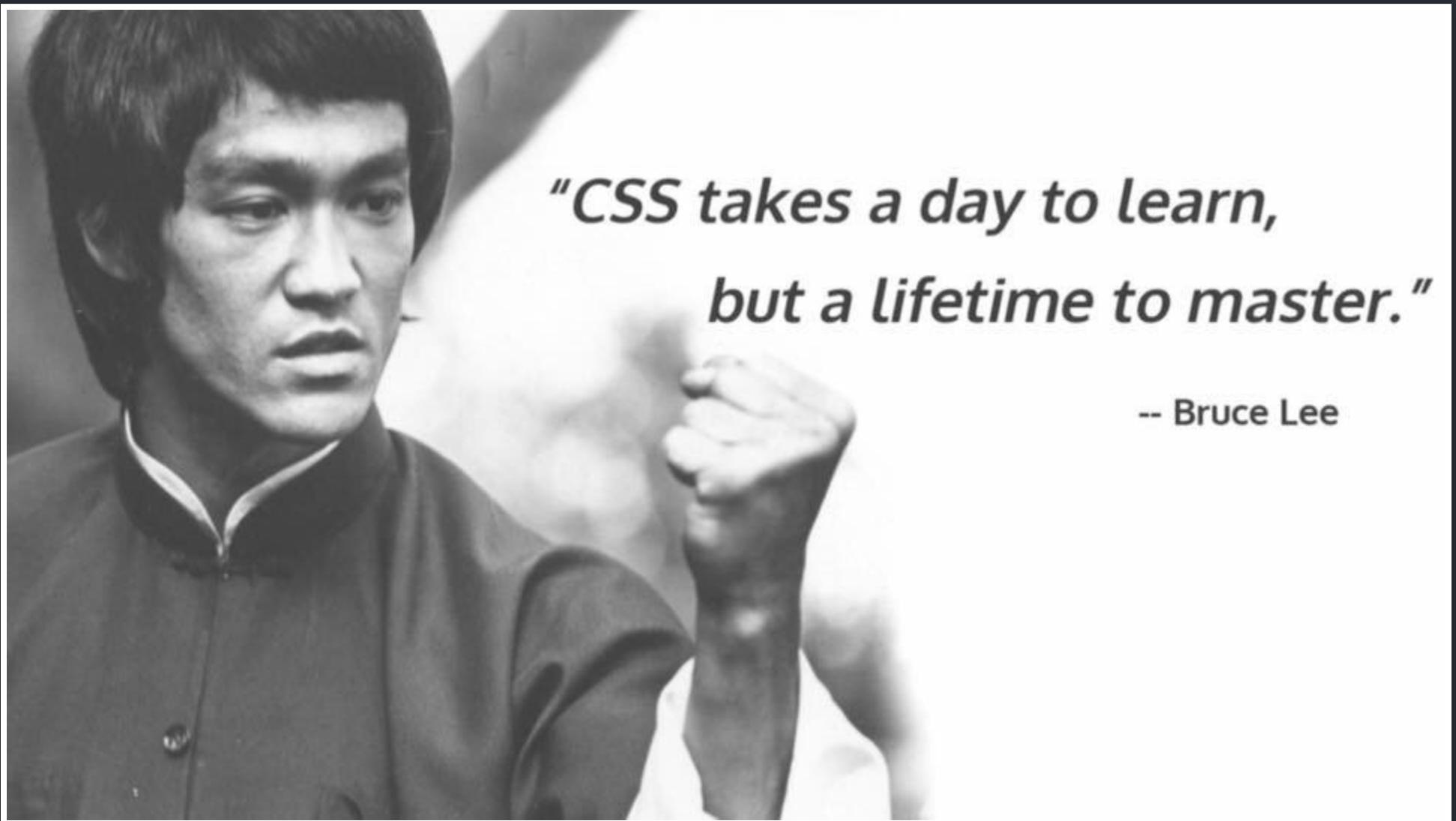
perspective.

BASTA!!1!

Make CSS great again!



Non stiamo parlando delle difficoltà del CSS



*"CSS takes a day to learn,
but a lifetime to master."*

-- Bruce Lee

A tal riguardo...

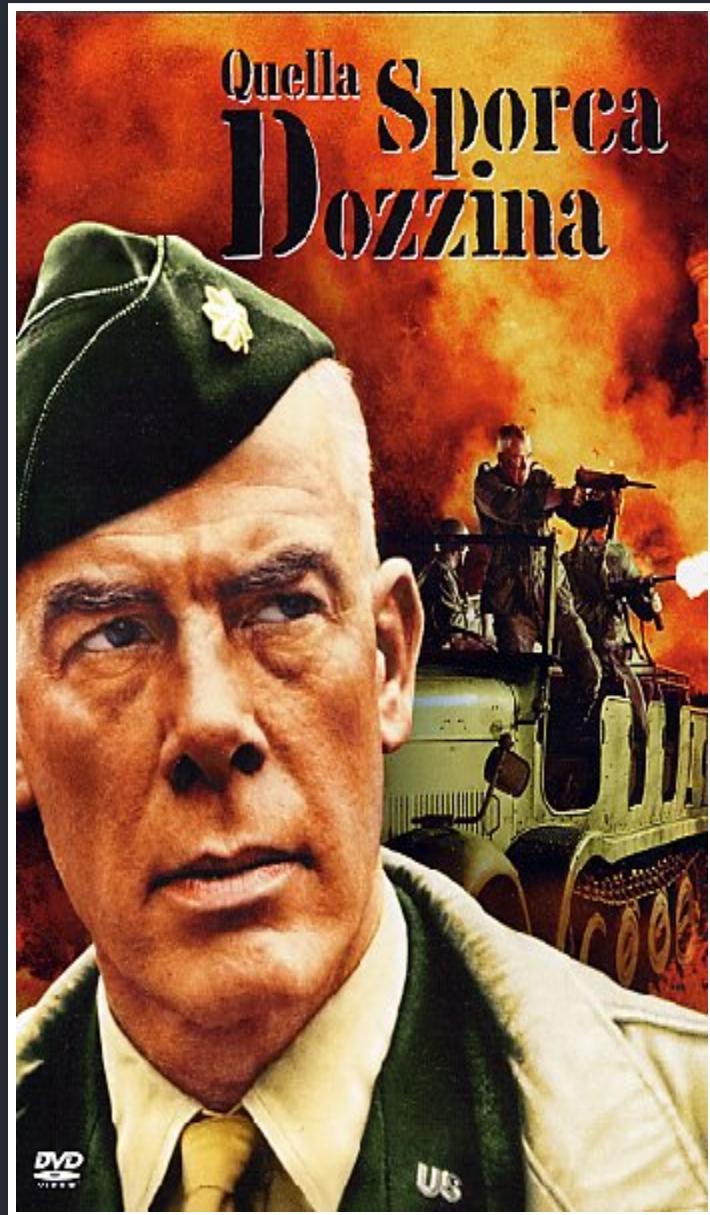


Quindi?

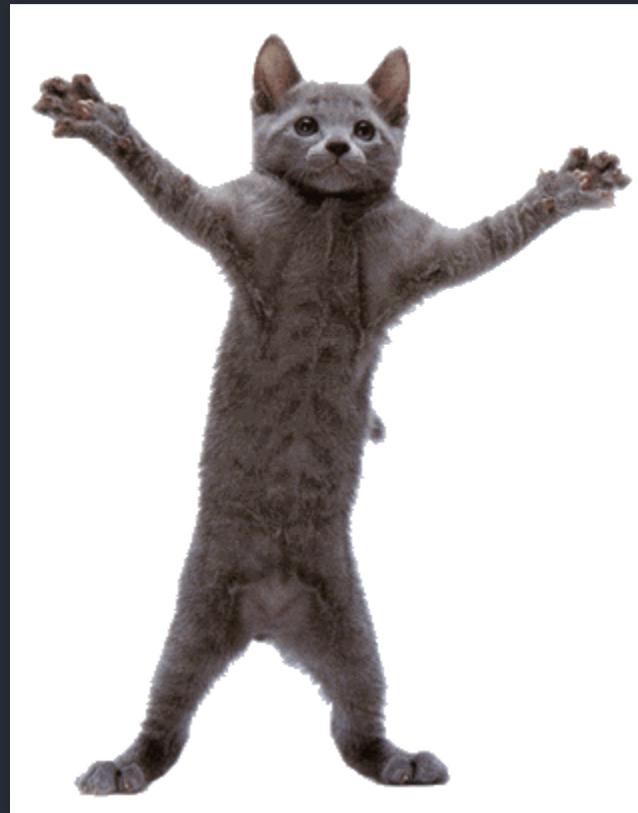


E il CSS è divertente!

Sì ma sto titolo?



Let's start meow!!!



AGAIN!

#1 :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!

The app is unavaible now!

In combo con :not

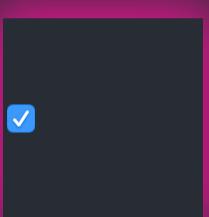
[Load content](#)



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

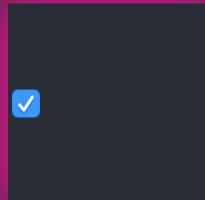
#2 :checked

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

```
<label for="checkbox2">Label</label>
<input type="checkbox" name="checkbox2" id="checkbox2" checked="">
```

Label



Rendiamolo interessante

Sibling combinator

sel ~ sel

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

Label



Demo

Ok, nella vita vera?

Custom checkbox!!!

```
<div class="cb">
  <div class="cb__cont">
    <input id="cbId1" type="checkbox" class="cb__input">
    <svg version="1.1" stroke-miterlimit="10" xmlns="http://www.w3.org/2000/svg">
      <path fill="transparent" stroke="#000000" stroke-width="4" stroke-linecap="square" d="M 10 10 L 10 20 L 20 20 L 20 10 Z"/>
    </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

Test Text

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus vulputate urna nulla, sit amet consequat dui eleifend non. Vestibulum dapibus interdum felis. Sed pellentesque placerat est non tristique. Sed lacinia tempor tortor et cursus. Fusce facilisis iaculis mi sit amet fermentum. Quisque consectetur nisl a libero porttitor tempus. Nunc feugiat consequat est, id interdum eros tincidunt eget. Phasellus condimentum ante ac tristique placerat. Proin sollicitudin nibh eget enim ultricies condimentum. Donec tristique ultrices laoreet. Donec et lacinia leo, non sodales massa. Nam libero elit, convallis et ornare sit amet, efficitur vel dolor. Suspendisse potenti.

Praesent tristique nunc in vestibulum finibus. Nullam orci leo, iaculis at nulla vitae, finibus imperdiet felis. Proin et arcu semper, imperdiet elit viverra, pretium velit. Donec sagittis eros tincidunt laoreet blandit. Duis eu nulla a tortor eleifend eleifend sed sed eros. Praesent scelerisque sapien vitae dui rutrum sagittis. Nullam blandit eros eu auctor accumsan. Fusce enim mauris, ultricies et metus dapibus, lobortis imperdiet erat. Nunc id diam elit. Nullam non tellus ut quam convallis scelerisque.

Test Text

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

Tab One *Tab Two* *Tab Three* *Tab Four*

This is Panel One

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris vel leo sem. Sed nec quam sit amet lorem volutpat ullamcorper ut sed massa. Sed vel felis velit. Nullam et turpis sed dui auctor vehicula quis a dui.

Aliquam vitae leo et sapien volutpat accumsan quis eget turpis. Etiam ac metus vitae purus semper pretium. Curabitur id nisl nisl. Cras ut massa sed dolor ullamcorper consequat ut sed dolor.

Nam sodales, mi eu convallis adipiscing, ligula justo consectetur tellus, tincidunt vestibulum tortor elit a augue. Ut elementum ultricies orci, vel luctus neque varius in. Phasellus turpis nunc, eleifend ac fringilla at, malesuada in eros.

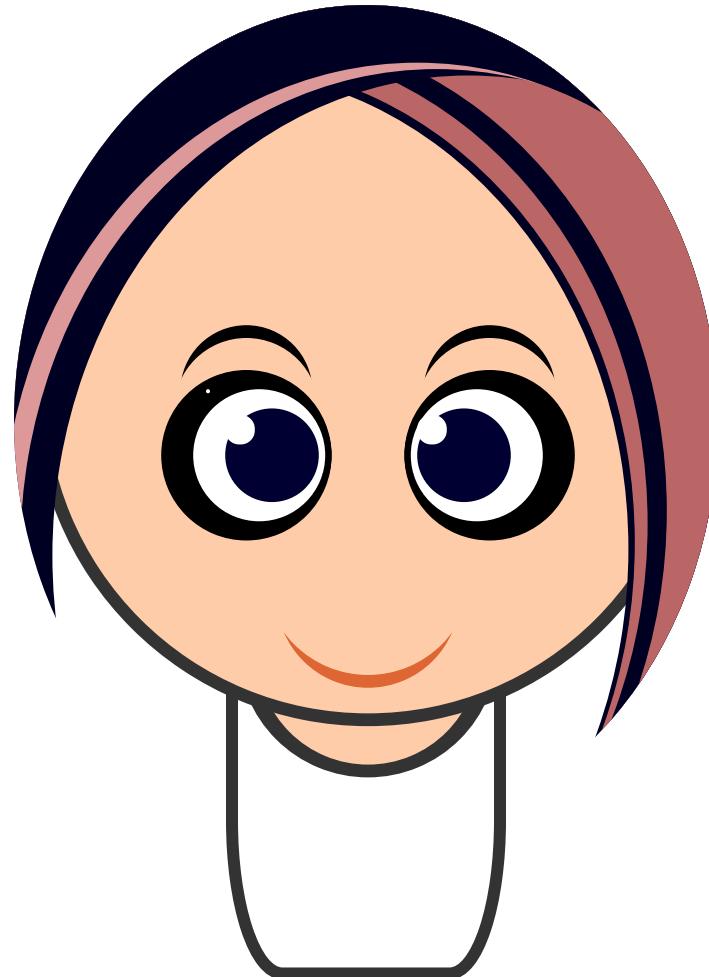
See the Pen [CSS Only Tabbed Content](#) by Stephen Greig (@stephengreig) on CodePen.

[HTML](#)[CSS](#)[Result](#)EDIT ON
 CODEPEN

One element girl

expression

color



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

#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;
}
```

Pippo Link

PIPPO!!1!

See the Pen [target example](#) by Davide Di Pumbo (@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!!!

[HTML](#)[CSS](#)[JS](#)[Result](#)

EDIT ON

Slide 1

1 2 3

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

CSS ONLY MODAL

Using :target you can do a pure CSS modal. Click the above button to see it in action.

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

All CSS Accordion

First Accordion

Second Accordion

Third Accordion

For multiple accordions on a single page, repeat process, but assign different id names. [Multiple accordions example](#)

Beware use on Android 2.2 and 2.3. The default browser on these versions does some curious things.

See the Pen [CSS Accordion](#) by pollardId (@pollardId) on CodePen.

#4 :placeholder-
shown

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

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

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

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

Placeholder

Un utilizzo reale?

[HTML](#)[SCSS](#)[JS](#)[Result](#)[EDIT ON
CODEPEN](#)

Sign up

The form consists of four input fields: 'First', 'Last', 'Email', and 'Password'. Each field has a label positioned above it and a placeholder text inside the input field. A 'Sign up' button is at the bottom. A small '89' badge is located on the right side of the form.

First Last

Email

Password

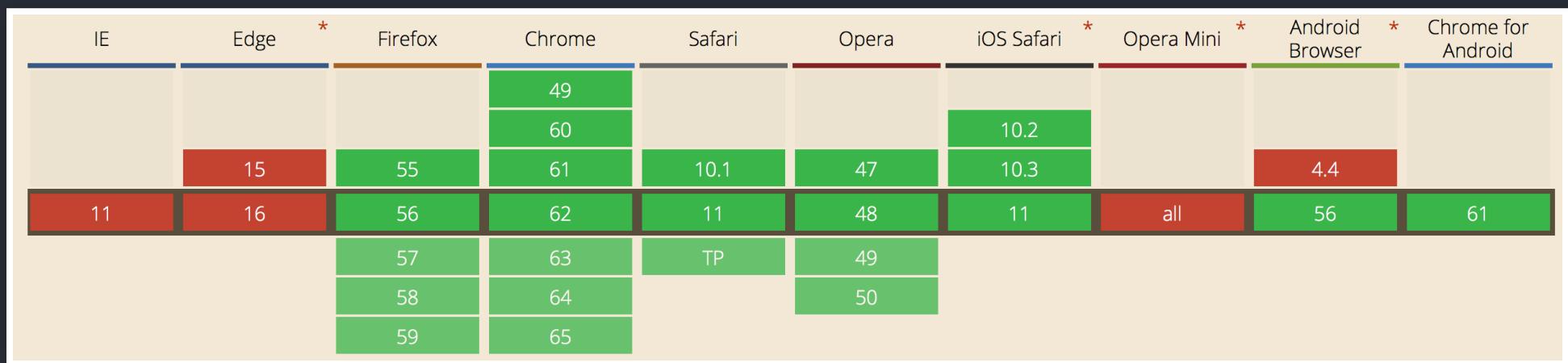
Sign up

89

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

Semplice facile e veloce,
no?

ma...



IE

Edge

*

Firefox

Chrome

11

15

16

55

56

49

60

61

62

57

58

59

63

64

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);
});
```

#5 :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

Enter a URL:

`http://url.it`

Enter an email address:

`info@info.it`

Enter a date:

`DD/MM/YYYY`

Submit

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

Enter a URL:

`http://url.it`

Enter an email address:

`info@info.it`

Enter a date:

`DD/MM/YYYY`

Submit

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

Cose negative?

No serio, io non le ho trovate 😊

#6 currentColor

È una variabile sempre valorizzata con il valore della proprietà **color** corrente

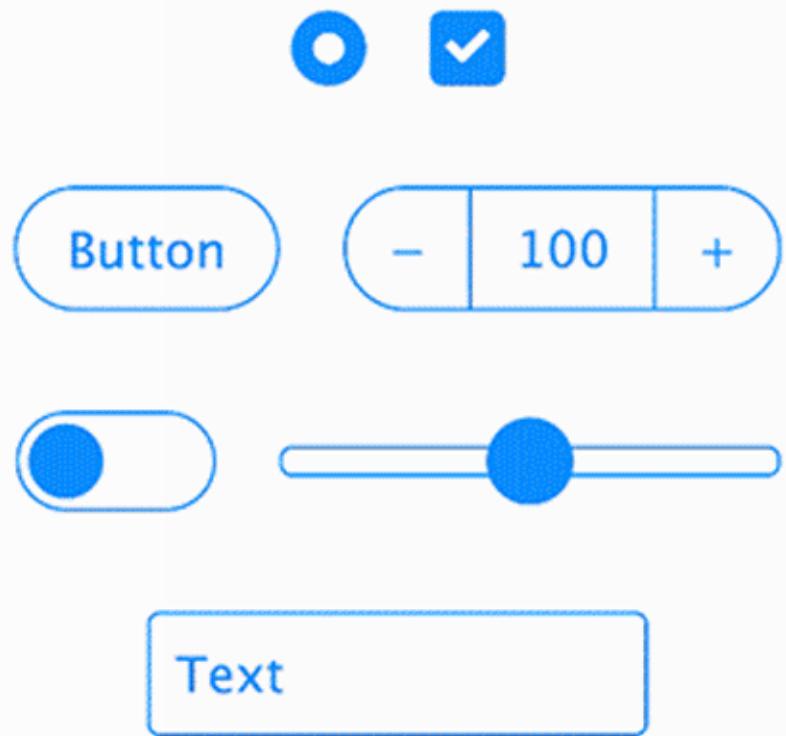
```
<div class="example-currentColor">Example</div>
<style>
  .example-currentColor{
    color: #E0168F;
    border-bottom: 0.5em solid currentColor;
    box-shadow: 0px 0px 1px currentColor;
  }
</style>
```

Example

Real life?



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



```
<!DOCTYPE html>
<html data-montage-skin class="montage">
  <head>...</head>
  <body>
    <div data-montage-id="main" class="montage">
      <sections></sections>
    </div>
  </body>
</html>
```

The screenshot shows a developer's interface with a code editor and a preview pane. The code editor displays the following HTML and CSS:

```
html.r
Styl.
element
}
html {
  &#x2022; for
  &#x2022; hei
  &#x2022; fil
  &#x2022; background-color: white;
  &#x2022; color: dodgerblue;
}
```

The preview pane shows a landscape image with a color gradient overlay. A color picker on the right shows a gradient from red to purple. A color selection dialog is open, showing a color swatch labeled "dodgerblue".

Il mio preferito è con le SVG!

Button ⓘ

Info ⓘ

Success ⓘ

Error ⓘ

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

Siamo solo a
metà

Acceleriamo!

#7 :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 +



A tool to help build Quantity Queries for your projects

WHAT IS A QUANTITY QUERY?

By [drewminns](#)

Follow me [@drewisthe](#)

[Tweet](#)

Build a query

WHICH ELEMENT WILL BE
COUNTED

ex. ul li

Your Code

Copy and paste the code below into your styles

```
// Build a query on the left
```

Quantity Queries

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;  
}
```



Broken image

#8 Broken
images

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

Confucio

```
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; }
```

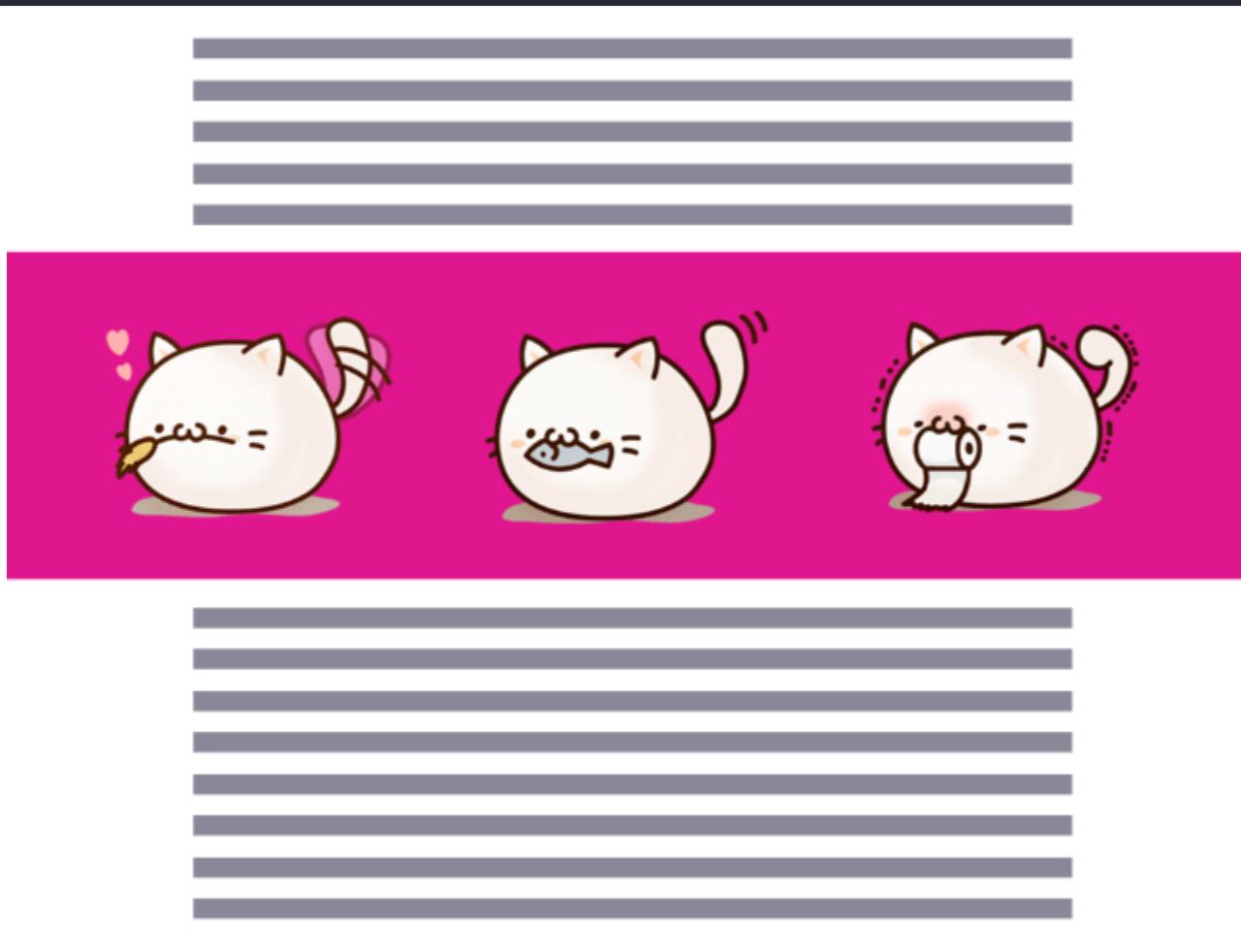
 Image not available:
Awesome stuff

Due cose:

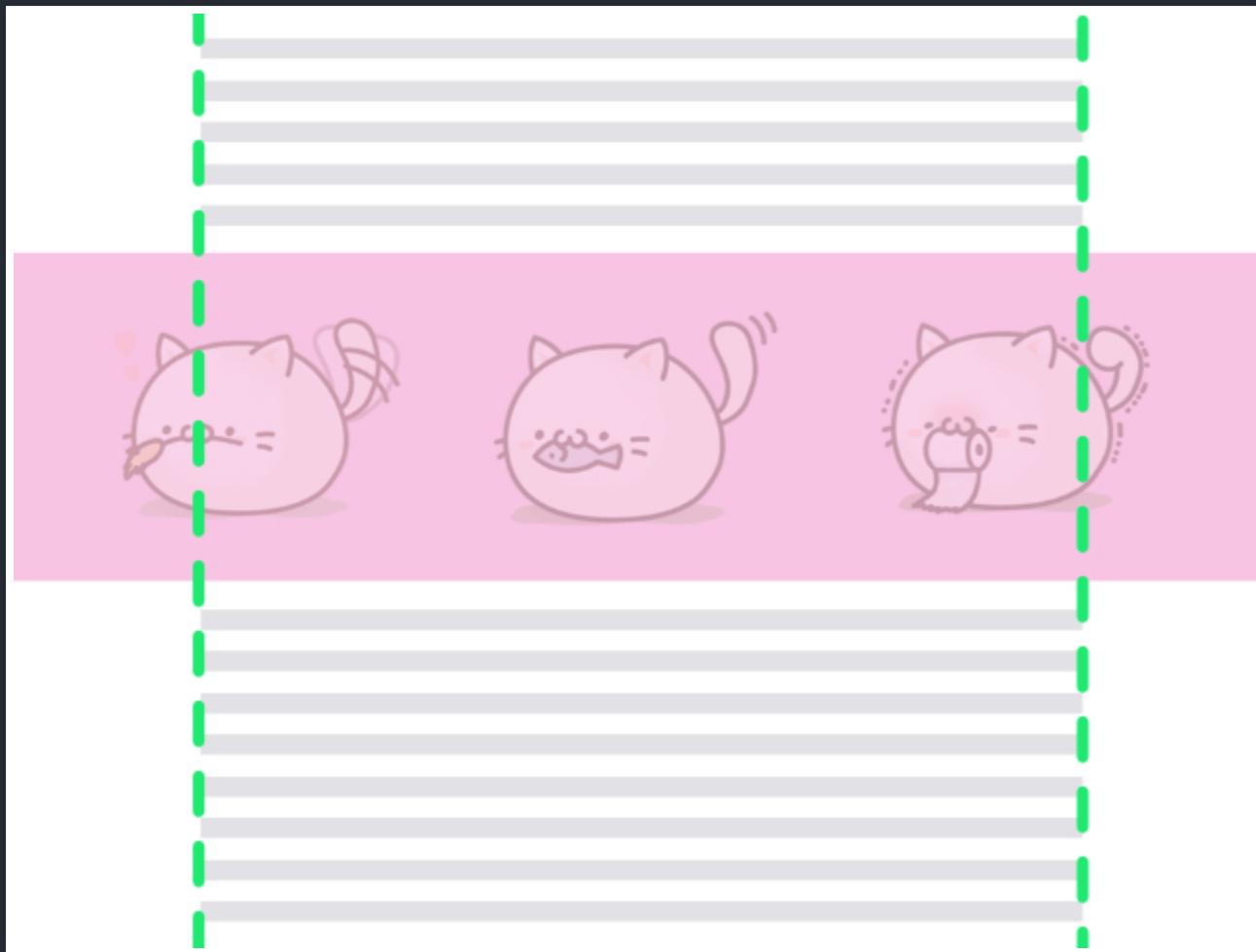
- `content: "..."; attr(alt);`
- Le brutte notizie

Sto trucco non funziona su IE e Safari...

#9 Escape the
cage



Cats are frome here



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. Morbi
    <div class="our-div"></div>
    Sed lacus dolor, blandit et tempor at, consectetur luctus velit.
  </div>
</div>
```

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.

```
.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;

#10 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="<br> [ #bootstrap col-sm-6 col-md-2 ]<br> [ #utility red big ]<br> my-selector<br>"></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!

#11 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="http://www.w3.org/2000/svg">
  <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 Result

EDIT ON 

```
.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 0.8s ease-in forwards 1.5s;  
}
```



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

Ma il mio esempio preferito?

- What a wonderful checkbox

#12 Bonus track

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

Pug SCSS Result EDIT ON CODEPEN

```
@each $element in (meta, link, br, hr) {
  #{$element}::after {
    content: '#{$element} element';
    display: block;
  }
}

head {
  display: block;
}

meta:first-of-type, link:first-of-type, br, hr {
  display: block;
  background: #E0168F;
  padding: 1rem;
  margin: 1rem;
  color: white;
  border: none;
}

body, head {
  background: #333;
  font-size: 2rem;
  font-family: sans-serif;
}
```

meta element

link element

hr element

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">
  ▼<span class="counter">
    ... ::before == $0
    </span>
    <span class="code">:empty
    </span>
  </h1>
</section>
```

... div section section h1 span ::before

Styles Computed Event Listeners ➞

Filter :hov .cls -

```
.counter::before { black.css:3
  content: "#" counter(trick);
}
```



THANK YOU