

به نام خدا

# EMMET



LinkedIn: Ali Moeinian

## فهرست مطالب

- مقدمه ❖
- Emmet چیست؟ ❖
- Emmet پلاگین اضافه کردن ❖
- DOCTYPE ❖
- Class, ID ❖
- Nesting ❖
- Sibling ❖
- Multiple ❖
- Lorem Ipsum ❖
- Text ❖
- Climb Up ❖
- Grouping ❖
- Attribute ❖
- Numbering ❖
- Input, a ❖
- Link ❖

**Margin, Padding** ❖

**منابع آموزشی** ❖

**Emmet Cheat Sheet** ❖

سلام !

بعد از تجربه‌ی موفقی که در خصوص آموزش Git و GitHub داشتم،  
این بار با هم Emmet را یاد می‌گیریم.

شما می‌توانید من رو در لینکدین دنبال کنید :

A screenshot of Ali Moeinian's LinkedIn profile. It features a circular profile picture of a young man with dark hair, wearing a grey jacket over a white shirt, set against a background of a city skyline under a clear blue sky. To the right of the profile picture is a large, blurred background image of a computer screen displaying a dense grid of colorful code snippets. At the top right of the profile page is a small circular icon with a blue pen nib. Below the profile picture, the name "Ali Moeinian" is displayed with a speaker icon indicating he is a public speaker. His bio reads: "Student of Computer Software Engineering Talks about #python, #programming, and #webdevelopment Esfahan, Iran · Contact info". Below his bio, it says "4,797 followers · 500+ connections". At the bottom of the profile section are three buttons: "Open to", "Add profile section", and "More". To the right of the profile area, there is a logo for Islamic Azad University and the text "Islamic Azad University".

امیدوارم این آموزش هم مثل فایل‌های قبلی مورد استقبال قرار بگیره.

## Emmet – 1 چیست و چرا باید Emmet را بلد باشیم ؟

یک پلاگین خیلی باحاله که سرعت توسعه‌ی وب رو میتونه  
چند برابر کنه.

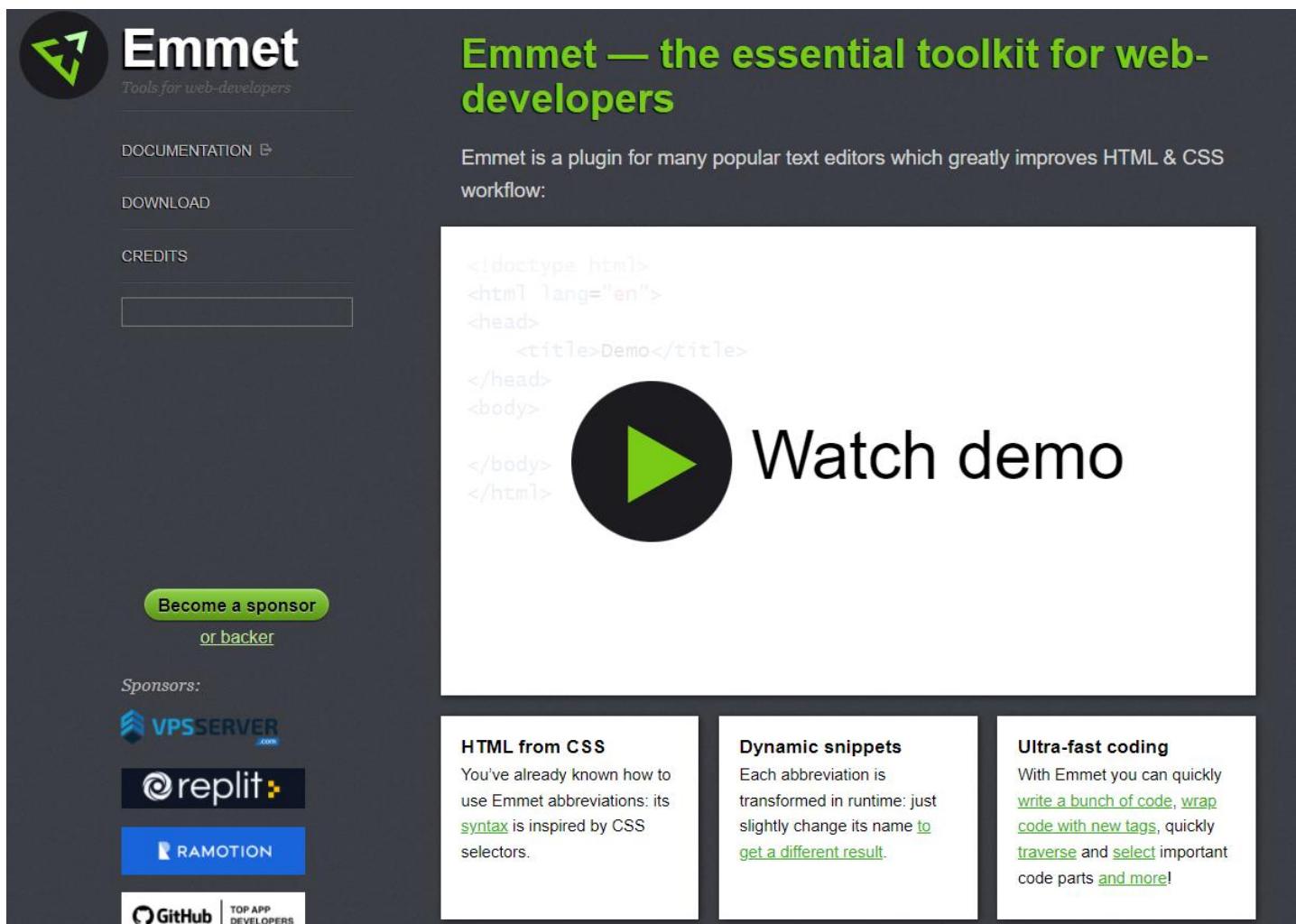
اگر با این پلاگین آشنا باشید، سرعت نوشتن فایل‌های HTML و CSS،  
حداقل 2 برابر میشه.

یادگیری این پلاگین خیلی راحته و اصلا هم سخت نیست.

امکان داره بر حسب تجربه شما از این پلاگین به طور روزانه استفاده  
کنید و خبر نداشته باشید 😊

در حقیقت خیلی از دولوپر‌ها اصطلاح Auto complete رو به کار  
میبرند، اما در واقعیت از قوانین و اختصار‌های Emmet دارند استفاده  
میکنند.

# وب سایت رسمی : Emmet



The screenshot shows the official website for Emmet. At the top left is the Emmet logo (a stylized 'E' icon) and the word 'Emmet'. Below it is the tagline 'Tools for web-developers'. On the left sidebar, there are links for 'DOCUMENTATION', 'DOWNLOAD', and 'CREDITS'. A green button says 'Become a sponsor or backer'. Below that is a section for 'Sponsors' with logos for VPS SERVER, repl.it, RAMOTION, and GitHub. The main content area has a title 'Emmet — the essential toolkit for web-developers' and a subtitle explaining it's a plugin for text editors. It includes a code snippet example and a large play button with the text 'Watch demo'. To the right of the play button are three boxes: 'HTML from CSS', 'Dynamic snippets', and 'Ultra-fast coding', each with a brief description.

Emmet — the essential toolkit for web-developers

Emmet is a plugin for many popular text editors which greatly improves HTML & CSS workflow:

```
<!doctype html>
<html lang="en">
<head>
  <title>Demo</title>
</head>
<body>
</body>
</html>
```

Watch demo

Become a sponsor  
or backer

Sponsors:

VPS SERVER

repl.it

RAMOTION

Github | TOP APP DEVELOPERS

**HTML from CSS**  
You've already known how to use Emmet abbreviations: its [syntax](#) is inspired by CSS selectors.

**Dynamic snippets**  
Each abbreviation is transformed in runtime: just slightly change its name [to get a different result](#).

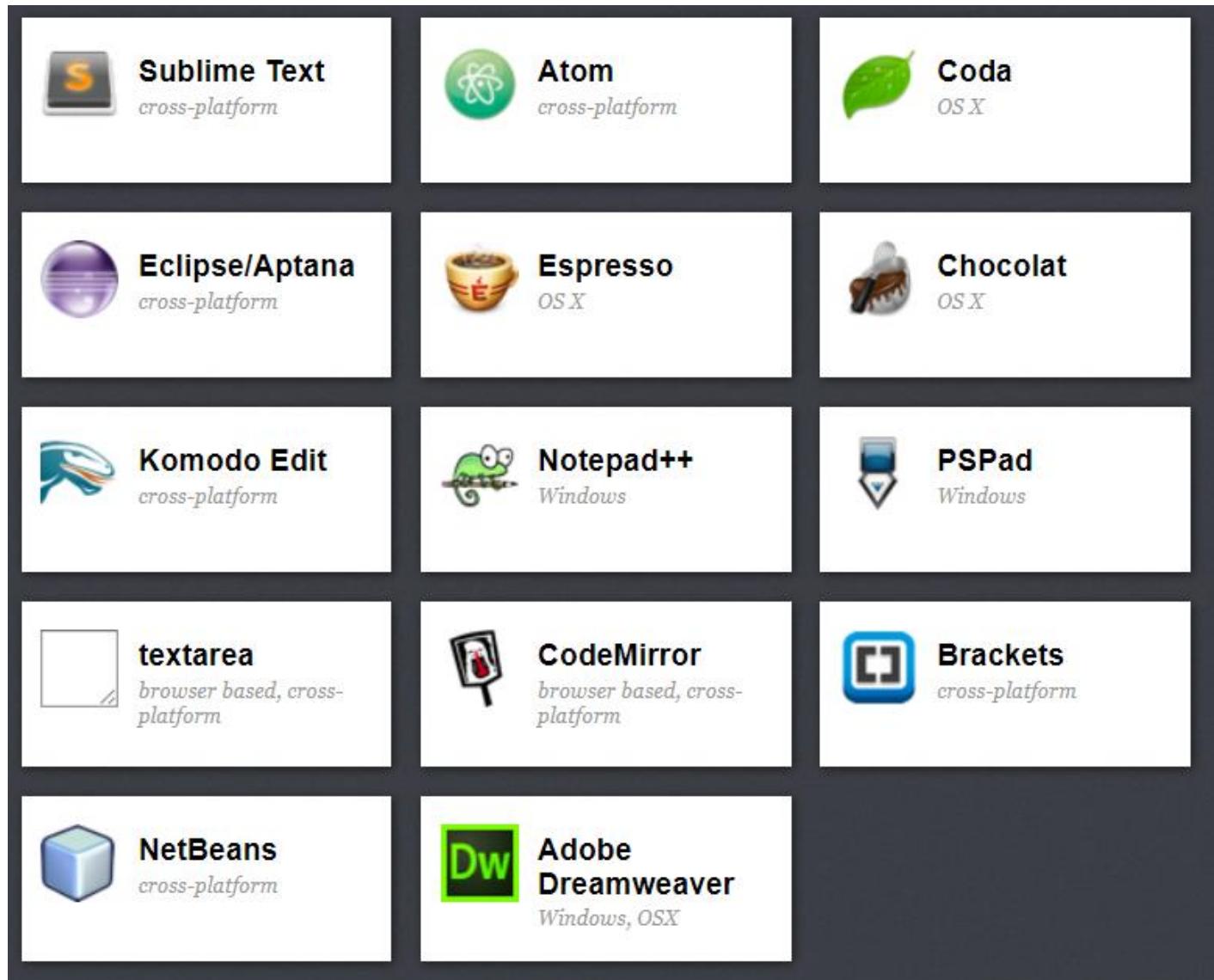
**Ultra-fast coding**  
With Emmet you can quickly [write a bunch of code](#), [wrap code with new tags](#), quickly [traverse](#) and [select](#) important code parts [and more!](#)

هر آنچیزی که نیاز دارید میتوانید در وبسایت Emmet پیدا کنید.

✖ داکیومنتش رو حتما مطالعه کنید ✖

## 2- اضافه کردن پلاگین IDE به ادیتور یا Emmet

همونطور که همه‌ی ماه‌ها میدونیم، پلاگین‌ها باید به محیط کار اضافه شوند. باید IDE‌های زیر دانلود کنید.



این پلاگین به طور پیش فرض بر روی ابزارهای زیر هست و نیازی به  
نصب نیست.

 <b>Visual Studio Code</b> <small>cross-platform</small>	 <b>SynWrite</b> <small>Windows</small>	 <b>WebStorm</b> <small>cross-platform</small>
 <b>PhpStorm</b> <small>cross-platform</small>	 <b>Vim</b> <small>cross-platform</small>	 <b>HTML-Kit</b> <small>Windows</small>
 <b>HippoEDIT</b> <small>Windows</small>	 <b>CodeLobster PHP Edition</b> <small>Windows</small>	 <b>TinyMCE</b> <small>browser based, cross-platform</small>
 <b>EverEdit</b> <small>Windows</small>	 <b>EditPlus</b> <small>Windows</small>	 <b>Emacs</b> <small>cross-platform</small>
 <b>WeBuilder</b> <small>Windows</small>	 <b>Oxygen XML Editor</b> <small>cross-platform</small>	

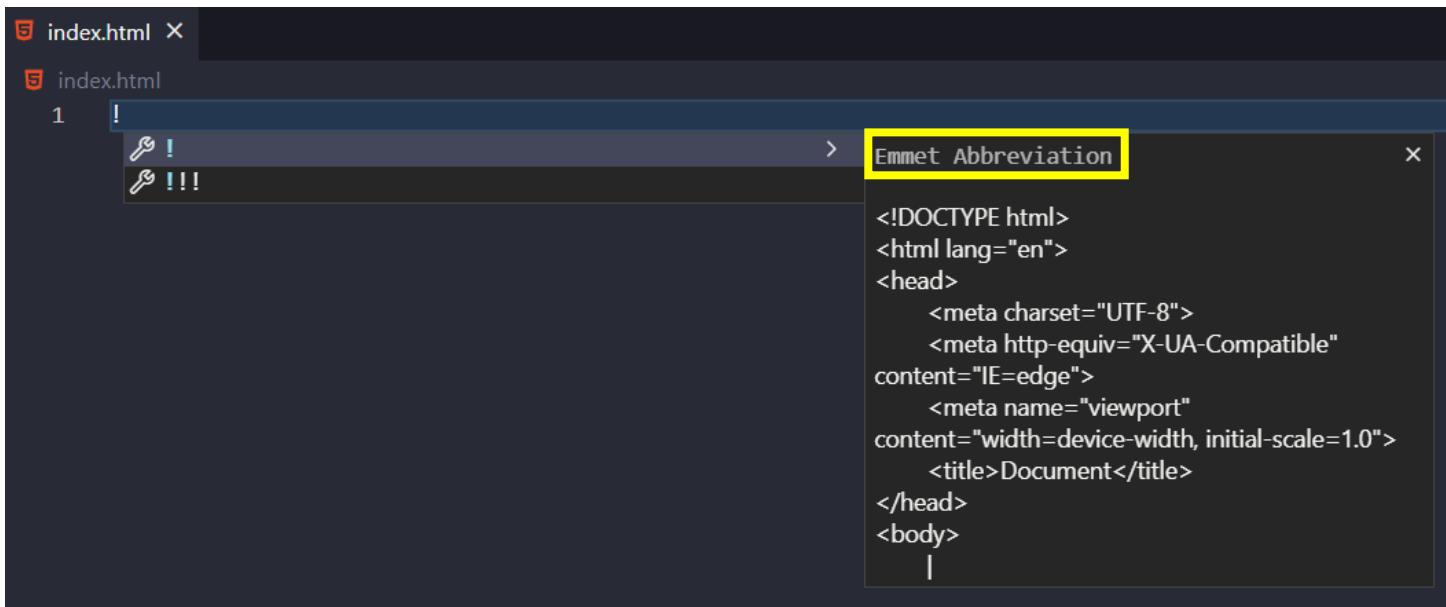
در این آموزش، ما از **visual studio code** استفاده میکنم.

# HTML



## DOCTYPE – 3

اول از همه یک فایل ساده‌ی HTML ساختم. به راحتی میتوانید با تایپ کردن علامت ! فایل رو کامل کنید.



The screenshot shows a code editor window with two tabs: 'index.html' and 'index.html'. In the first tab, at line 1, there is a cursor after a single exclamation mark ('!'). A dropdown menu titled 'Emmet Abbreviation' is open, displaying a template for an HTML document. The template includes the DOCTYPE declaration, HTML and head tags, meta charset and http-equiv tags, a title, and a body tag. The entire template is highlighted with a yellow box.

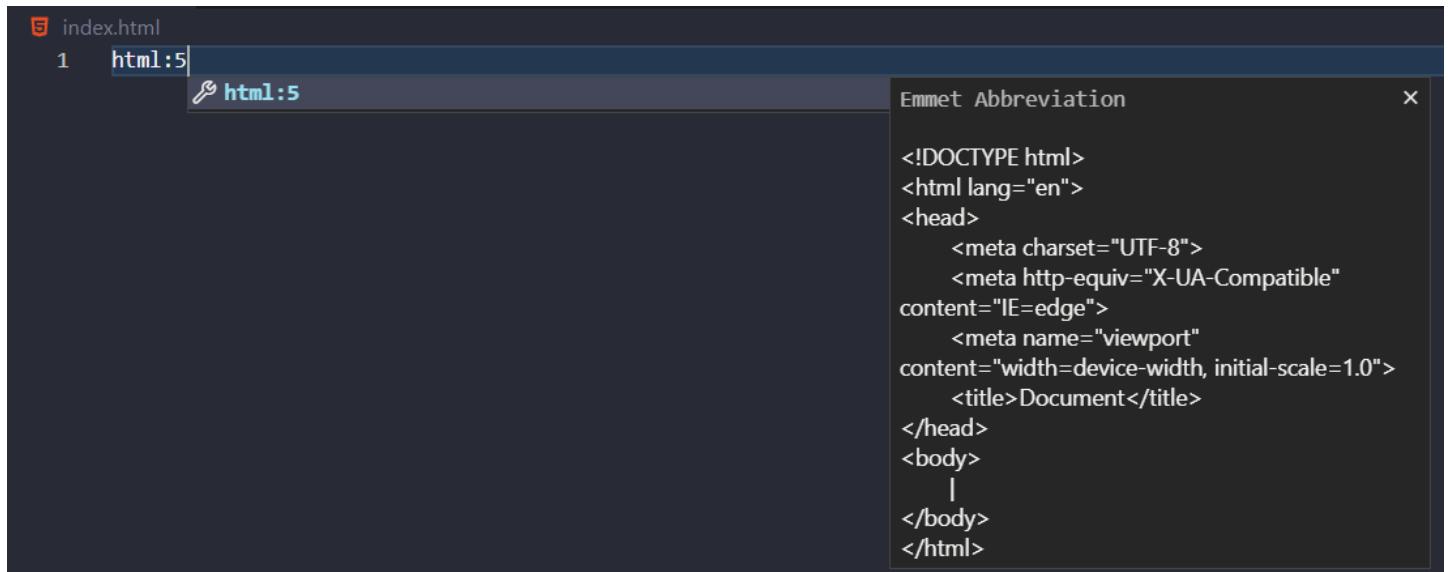


The screenshot shows the expanded HTML code generated by the Emmet abbreviation. The code includes the DOCTYPE declaration, HTML tag with lang="en", head tag containing meta charset="UTF-8", meta http-equiv="X-UA-Compatible" with content="IE=edge", meta name="viewport" with content="width=device-width, initial-scale=1.0", and a title tag with the text 'Document'. The code is numbered from 1 to 12.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  |   <title>Document</title>
8  </head>
9  <body>
10 |
11 </body>
12 </html>
```

و این هم نتیجه

برای تکیمل Doctype HTML یک راه دیگر هم وجود دارد :



The screenshot shows a code editor interface with a dark theme. In the top left, there's a file icon and the text "index.html". Below it, the number "1" and the text "html:5" are visible. A dropdown menu titled "Emmet Abbreviation" is open on the right, containing the following code snippet:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    |
</body>
</html>
```

پس برای ایجاد کدهای پایه‌ی html5 دو راه وجود دارد :

1) استفاده از نماد !

html:5 (2

## Class, ID – 4

قطعا هر وب دولوپری کاربرد class و id را به خوبی میدونه.

به جای اینکه عین قبل، اول تگ مد نظر رو بسازید و بعد بهش کلاس بدید، Emmet کار رو ساده کرده برآتون و میتوانید از فرمول زیر استفاده کنید :

### TagName.ClassName



به مثال پایین توجه کنید

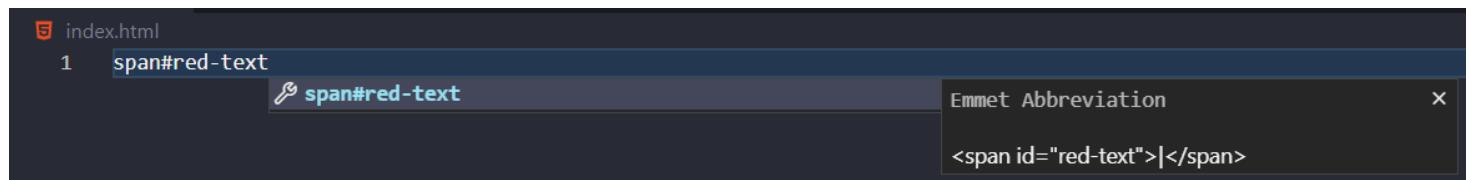
The screenshot shows a code editor window with a dark theme. In the top left, there's a file icon followed by "index.html". Below it, two lines of code are visible: line 1 starts with a comment "-- برای ساخت کلاس <!>" and line 2 contains the text "div.picture-container". To the right of the editor, a tooltip-like box appears with the title "Emmet Abbreviation". Inside the box, the text "<div class="picture-container">|</div>" is shown, indicating the expanded form of the abbreviation.

The screenshot shows a code editor window with a dark theme. At the top, it says "index.html > div.picture-container". Below, three lines of code are displayed: line 1 shows the opening tag "<div class="picture-container">" with the "class" attribute highlighted in green; line 2 is an empty line; line 3 shows the closing tag "</div>". To the right of the editor, there is a hand icon pointing right with the text "و نتیجه" next to it.

برای دادن آی دی هم باید از فرمول زیر استفاده کنید :

**TagName#ID**

به مثال زیر توجه کنید



```
index.html
1 span#red-text
```

Emmet Abbreviation

```
<span id="red-text">|</span>
```

نتیجه



```
index.html > span#red-text
1 <span id="red-text">
2
3 </span>
```

ما میتوانیم هم زمان هم از کلاس و هم از آی دی استفاده کنیم.

**فرمول : TagName#ID.Class**

میتوانید اول کلاس بدید و بعد آی دی؛ تفاوتی نداره.



مثال

A screenshot of the VS Code interface showing the Emmet Abbreviation feature. The code editor shows the input: `div.picture-container#picture-one`. A tooltip labeled "Emmet Abbreviation" displays the expanded code: `<div class="picture-container" id="picture-one">|</div>`.



نتیجه

A screenshot of the VS Code interface showing the result of the Emmet abbreviation expansion. The code editor shows the expanded code: `<div class="picture-container" id="picture-one">|</div>`.

اما باز هم Emmet کار مارو راحت تر کرده.

اگر صرفا میخواهید به یک div، کلاس یا آی دی بدمد، نیاز نیست که

بنویسید `.div#ID` یا `div.Class`

برای دادن کلاس به div کافیه بنویسید : `.Class`

برای دادن آی دی به div کافیه بنویسید : `#id`

A screenshot of the VS Code interface showing the Emmet Abbreviation feature. The code editor shows the input: `.container`. A tooltip labeled "Emmet Abbreviation" displays the expanded code: `<div class="container">|</div>`.





A screenshot of a code editor window titled "index.html". In the code area, the text "#container" is typed. A tooltip labeled "Emmet Abbreviation" appears above the cursor, containing the expanded code "<div id="container">|</div>".



## Nesting -5

در اکثر اوقات، از تگ های تو در تو خیلی زیاد استفاده میکنیم.

علی الخصوص زمانی که توجه زیادی روی semantic html داشته باشیم.

برای نوشتتن تگ های تو در تو با استفاده از Emmet میتوانیم از فرمول

زیر کمک بگیریم :

Father>child> ....

مثال

A screenshot of a code editor showing the Emmet Abbreviation panel. The main area shows the code: `1 section>ul>li`. A dropdown menu lists various Emmet abbreviations starting with `section>ul>li`, including `link:atom`, `link:css`, `link:favicon`, `link:im`, `link:import`, and `link:manifest`. The right panel displays the expanded Emmet abbreviation: `<section><ul><li>|</li></ul></section>`.

A screenshot of a code editor showing the expanded Emmet abbreviation. The code is: `1 <section> 2 <ul> 3 | <li></li> 4 </ul> 5 </section>`. The third line, containing the placeholder `| <li></li>`, is highlighted with a blue rectangle.

نتیجه

یادمون نمیره که میتوانیم قوانین Emmet رو همیشه با هم ترکیب

کنیم.

مثال

A screenshot of a code editor showing the Emmet Abbreviation panel. The main area shows the code: `1 section.menu-container>ul#food-list>li`. A dropdown menu lists various Emmet abbreviations starting with `section.menu-container>ul#food-list>li`, including `link:atom`, `link:css`, `link:favicon`, `link:im`, `link:import`, and `link:manifest`. The right panel displays the expanded Emmet abbreviation: `<section class="menu-container"><ul id="food-list"><li>|</li></ul></section>`.

نتیجه

A screenshot of a code editor showing the expanded Emmet abbreviation. The code is: `1 <section class="menu-container"> 2 <ul id="food-list"> 3 | <li></li> 4 </ul> 5 </section>`. The third line, containing the placeholder `| <li></li>`, is highlighted with a blue rectangle.

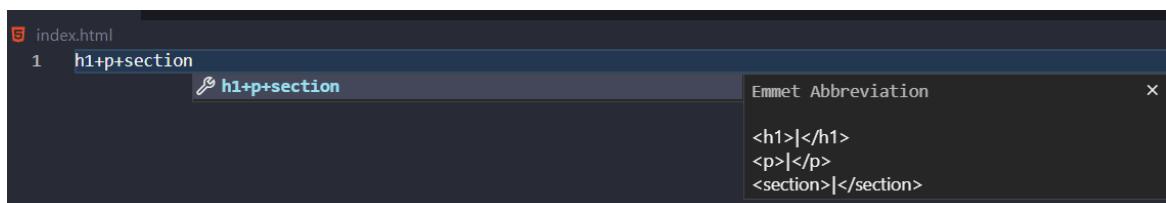
## sibling – 6

در قسمت قبل، ما یادگرفتیم چطور میتوانیم تگ های تو در تو بسازیم.

سوالی که اینجا مطرح میشے، اینه که الزاما همیشه تگ های ما، تو در تو نیستند.

حالا باید چیکار کنیم ؟

Tag1 + Tag 2 + .....



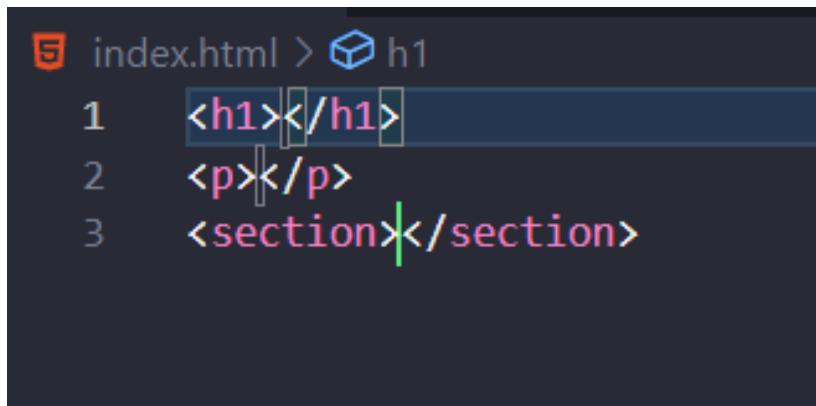
index.html

1 h1+p+section

Emmet Abbreviation

- <h1>|</h1>
- <p>|</p>
- <section>|</section>

مثال



index.html > h1

1 <h1></h1>

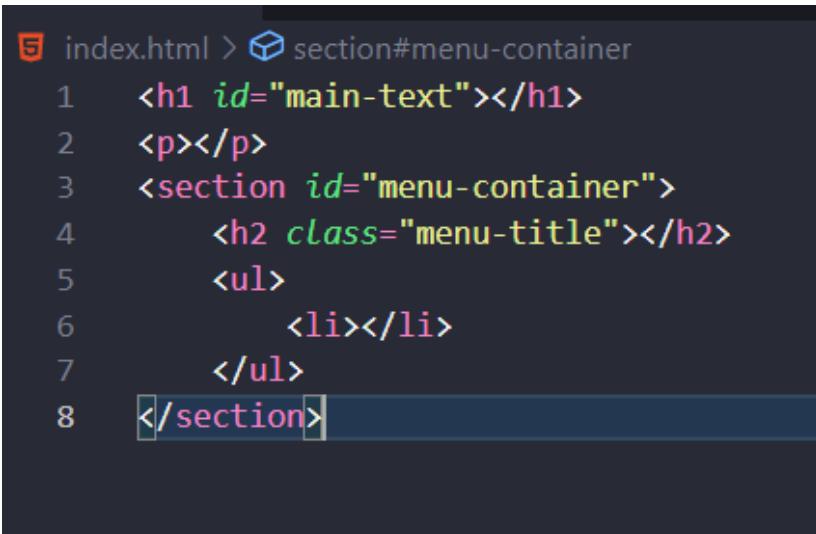
2 <p></p>

3 <section></section>

نتیجه

میتوانیم مطالبی که الان بلديم رو بيشتر با هم ترکيب کنيم.

ما اين قطعه کد رو ميخواهيم با استفاده از Emmet بنويسيم :



The screenshot shows a code editor window with the file path "index.html". The code is a template for a menu structure:

```
index.html > section#menu-container
1   <h1 id="main-text"></h1>
2   <p></p>
3   <section id="menu-container">
4       <h2 class="menu-title"></h2>
5       <ul>
6           <li></li>
7       </ul>
8   </section>
```

The code editor highlights the closing tag of the inner section element (line 8) in blue, indicating it is the target for expansion.

: Emmet قطعه کد بالا به وسیله‌ی



The screenshot shows a code editor window with the file path "index.html". The code is the Emmet abbreviation for the menu template shown above:

```
index.html
1   h1#main-text+p+section#menu-container>h2.menu-title+ul>li
2
```

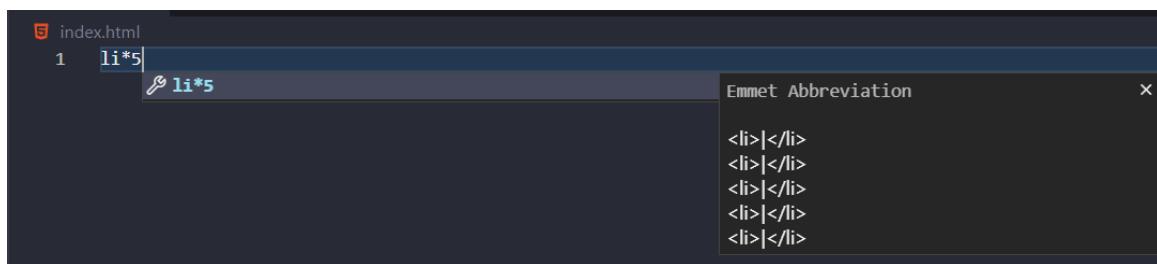
## Multiple – 7

خیلی وقت ها هست که نیاز داریم تا از یک تگ، چندتا داشته باشیم.

با توجه به قوانین Emmet، این کار رو میتوانیم با فرمول زیر انجام

بدیم :

### TagName\*Number



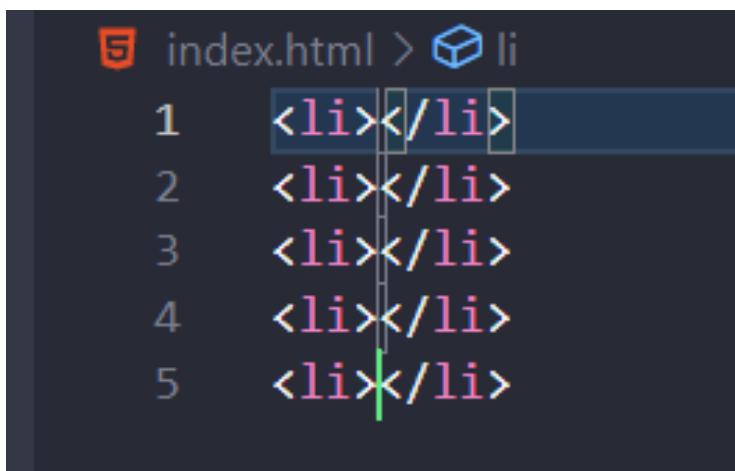
index.html

1 li\*5

Emmet Abbreviation

- <li>|</li>
- <li>|</li>
- <li>|</li>
- <li>|</li>
- <li>|</li>

مثال



index.html > li

1 <li></li>

2 <li></li>

3 <li></li>

4 <li></li>

5 <li></li>

نتیجه

# وقت ترکیب کردن قوانین رسید دوباره :

The screenshot shows a code editor interface with a dark theme. On the left, there is a file navigation bar with a file named "index.html". The main area displays the following HTML code:

```
1 section>div.menu-container>h2#menu-title+ul>li*5
```

A cursor is positioned at the end of the "li\*5" placeholder. A tooltip labeled "Emmet Abbreviation" appears on the right side of the screen, containing the expanded HTML structure:

```
<section>
  <div class="menu-container">
    <h2 id="menu-title">|</h2>
    <ul>
      <li>|</li>
      <li>|</li>
      <li>|</li>
      <li>|</li>
      <li>|</li>
    </ul>
  </div>
</section>
```

The screenshot shows a code editor interface with a dark theme. On the left, there is a file navigation bar with a file named "index.html". The main area displays the following HTML code:

```
1 <section>
  2   <div class="menu-container">
  3     <h2 id="menu-title">|</h2>
  4     <ul>
  5       <li>|</li>
  6       <li>|</li>
  7       <li>|</li>
  8       <li>|</li>
  9       <li>|</li>
 10     </ul>
 11   </div>
 12 </section>
```

The cursor is at the end of the "menu-title" placeholder in line 3. The code editor's interface includes syntax highlighting and a vertical line indicating the current position.



# مثل یوتوب تبلیغ بزارم این وسط

من در اکانت لینکدینم سعی میکنم اکثر وقت ها مطالب مفیدی منتشر

کنم و امیدوارم این مطالب کاربردی باشه برآتون.

Ali Moeinian  
Student of Computer Software Engineering  
3mo • Edited •

سلام به همه  
بالآخره فایل گیت که قولشو دادم آماده شد  
تا یک ماه پیش از گیت و گیت هاب هیچی نمیدونستم و واقعاً مونده بود من جطوری باید  
اگر مثل من (ورزن یک ماه پیش من) گیت رو بلد نیستید و دوست دارید یاد بگیرید این فایل  
خوبی متفاوتی هستون کار کرد  
اول از نصب گیت و تحت گیت بردن یک پروژه شروع میکنم و دستورات اصلیش رو با هم  
بررسی میکنم و ۳ تا مبتنی پروژه هم داخل این فایل ها آنجام دادم  
اشارة کردم (خواستم با رضایت خودتون باشه) که کار با گیت **Mohammad Taheri** به بست  
هاب رو آموزش میدن و درنهایت هم رفتم سراغ جادی که اسپ و لینک گیت هامون رو به  
ربیوزنیوری مربوط به جادی اضافه کردیم و بول ریکوئست زدیم و کربیی آخر  
قسمت دوم این فایل خلاصه ای از کورس مکسیمیلیان شوازرمولر خواهد بود که سعی  
میکنم زود آماده پیشه  
اگر تازه این پست من رو دیدید، لطفاً پست های جدید تر رو چک کنید تا در جریان فایل های  
ایدیت تر قرار بگیرد  
[#github](#) [#git](#) [#gitlab](#)



Ali Moeinian  
Student of Computer Software Engineering  
1mo • Edited •

سلام یک هفته میشه که دارم کورس از مکسیمیلیان (یودمی) رو میبینم.  
تکات خیلی باحال رو میگه که برای من تازه کار جذابه.  
فایل pdf مطالشون رو هم داخل گیت هایشون گذاشتند، گفتم شاید جالب باشه برآتون 😊  
آدرس گیت هاب <https://lnkd.in/d3x7wAfx>

## Clean Code

This document contains a summary of the concepts taught throughout the course. Use it in conjunction with the course lectures.

(c) 2020 Academind GmbH / Maximilian Schwarzmüller

Ali Moeinian  
Student of Computer Software Engineering  
4mo • Edited •

فایل کامل 21 قسمت جالش پایتون 267 صفحه 🎉 خودم باورم نمیشه اینقدر شد!  
تازه نکته ای جالشی اینجاست من سعی کدم فقط مطالب کلی و تیتر های مهم رو دوره کنم  
و اصلاً وارد میاخت دیگه مثلاً regex و Generator ها و .... نشدم 😊

ملون از همه ❤️🌟

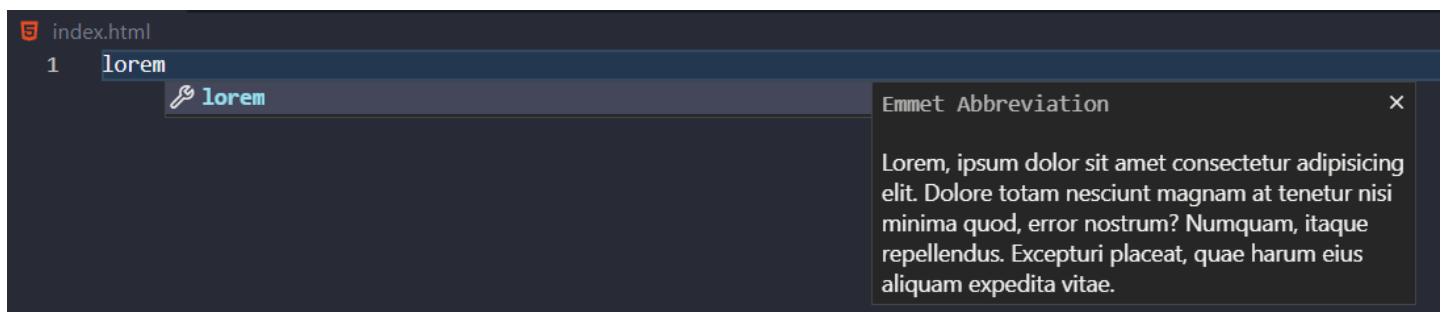
#جالش\_21\_روز\_پایتون

A slide titled "21 day challenge with Python". It features a large Python logo in the center. Below the logo, there is a list of 6 items: 1-Python History, 2-Python source, 3-Python IDE, 4-Python text editor, 5-Python courses, and 6-Python documents. At the bottom left, there is some sample Python code: `Print("what do we need ?".title())`.

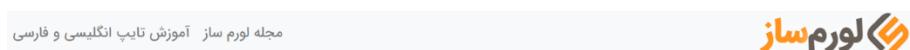
## Lorem Ipsum – 8

قطعاً تاحالاً از متن ساختگی معروف به لورم ایپسوم استفاده کردید.

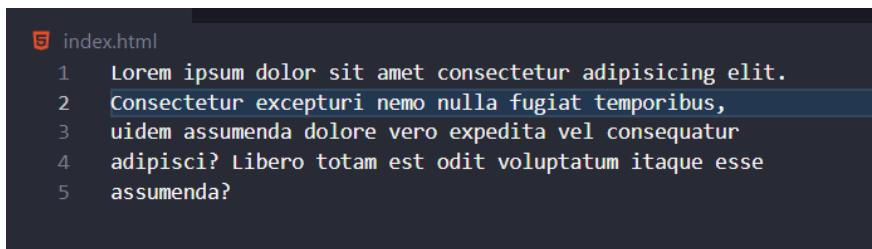
جالب‌ه بدونید که این متن ساختگی، از جمله ویژگی‌های Emmet هست.



از این متن برای تست کردن سایز فونت، نوع فونت رنگ و ... استفاده می‌شود و نسخه‌ی فارسی هم دارد:

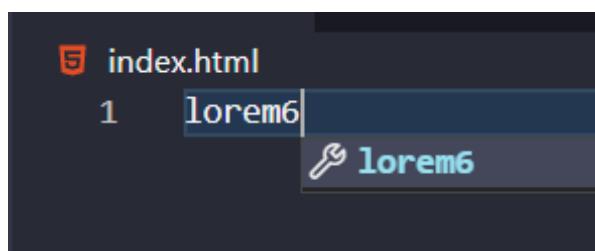


ممکنه این متن آماده رو بخواهید به صورت کوتاه تر استفاده کنید که میتوانید از فرمول روبرو استفاده کنید :



```
index.html
1 Lorem ipsum dolor sit amet consectetur adipisicing elit.
2 Consectetur excepturi nemo nulla fugiat temporibus,
3 uidem assumenda dolore vero expedita vel consequatur
4 adipisci? Libero totam est odit voluptatum itaque esse
5 assumenda?
```

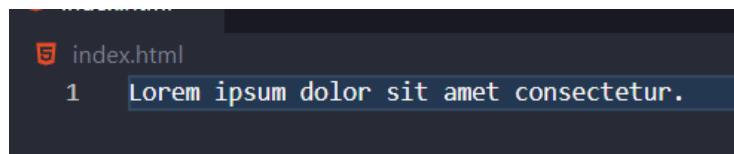
متن اصلی



```
index.html
1 lorem6
```

A tooltip shows the expansion of 'lorem6': `lorem6`

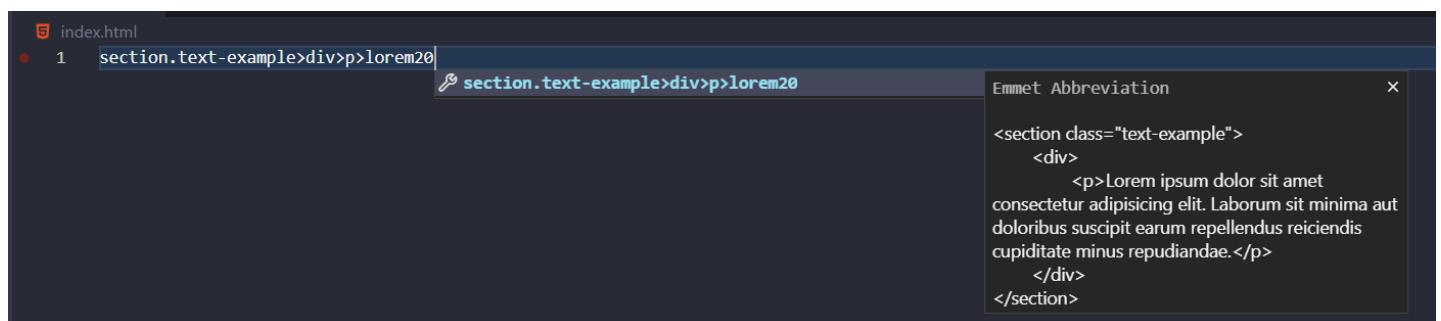
اعمال محدودیت



```
index.html
1 Lorem ipsum dolor sit amet consectetur.
```

نتیجه

وقت ترکیب کردن رسید



```
index.html
1 section.text-example>div>p>lorem20
```

The Emmet Abbreviation panel shows the expanded code:

```
<section class="text-example">
  <div>
    <p>Lorem ipsum dolor sit amet
    consectetur adipisicing elit. Laborum sit minima aut
    doloribus suscipit earum repellendus reiciendis
    cupiditate minus repudiandae.</p>
  </div>
</section>
```

```
index.html > section.text-example
1  <section class="text-example">
2    <div>
3      <p>Lorem ipsum dolor sit amet consectetur, adipisciing elit. Sed dolor ducimus, suscipit architecto quis quae corporis saepe sapiente possimus eos.</p>
4    </div>
5  </section>
```

## Text – 9

همونطور که دیدیم، Emmet صرفا برای ما تگ نمیسازه.

میتوانیم محتویات اون تگ ها رو هم تعیین کنیم، دقیقا عین مثال های بالا.

برای اضافه کردن متن دلخواه، باید از فرمول زیر استفاده کنیم :

**TagName { Text }**

نکته :

وقتی تعداد تگ ها خیلی میره بالا، مثل منو ، بهتره از این روش

استفاده نکنید چون واقعا خیلی کد شلوغ میشه.



مثال

index.html

```
1 div.menu-container>h2{Menu}>ul>li*5
```

Emmet Abbreviation

```
<div class="menu-container">
  <h2>Menu</h2>
  <ul>
    <li>|</li>
    <li>|</li>
    <li>|</li>
    <li>|</li>
    <li>|</li>
  </ul>
</div>
```

index.html

```
1 <div class="menu-container">
2   <h2>Menu</h2>
3   <ul>
4     <li>|</li>
5     <li>|</li>
6     <li>|</li>
7     <li>|</li>
8     <li>|</li>
9   </ul>
10 </div>
```



نتیجه

index.html

```
1 a[contact us]
```

Emmet Abbreviation

```
<a href="#">contact us</a>
```



مثال

index.html

```
1 <a href="#">contact us</a>
```



نتیجه

با قوانین جالبی که تا حالا یاد گرفتیم، میتوانیم الان یک فرم خیلی ساده

طراحی کنیم :

```
index.html
1  section>div.form-container>form>h2{This is a form}+input*4+button{submit}
```

```
index.html > section
1  <section>
2      <div class="form-container">
3          <form action="">
4              <h2>This is a form</h2>
5              <input type="text">
6              <input type="text">
7              <input type="text">
8              <input type="text">
9              <button>Submit</button>
10         </form>
11     </div>
12 </section>
```

## Climb Up – 10

میخواهیم تکه کد زیر رو به شیوه Emmet بنویسیم :

```
index.html > section
1  <section>
2    <div>
3      <form action="">
4        <input type="text">
5      </form>
6      <h1>Form here :</h1>
7    </div>
8  </section>
```

دقت کنید که تگ h1 بیرون از form قرار دارد.

```
section>div>form>input+h1
```

راه حل

```
<section>
  <div>
    <form action="">
      <input type="text">
      <h1></h1>
    </form>
  </div>
</section>
```

نتیجه

راه حل ما مشکل داره !

اگر دقت کنید، تگ h1 بیرون از form نیست.

خب باید چیکار کنیم؟

باید از علامت ^ استفاده کنیم و این درس جدید این قسمت بود.

```
section>div>form>input^h1{Form here :}
```

راه حل

```
<section>
  <div>
    <form action="">
      <input type="text">
    </form>
    <h1>Form here :</h1>
  </div>
</section>
```

نتیجه



دقیقا همونی شد که میخواستیم

نکته‌ی اصلی اینه که شما با استفاده از `^` تگ مورد نظرتون رو یک سطح عقب تر میفرستید، اگر `2` تا `2` استفاده کنید، به `2` سطح عقب تر و الی آخر.

مثلا در این قسمت، تگ `h1` رو هم سطح با تگ `section` قرار دادم :



```
section>div>form>input^^^h1{Form here :}
```

```
<section>
  <div>
    <form action="">
      <input type="text">
    </form>
  </div>
</section>
<h1>Form here :</h1>
```

بعضی وقت ها هست، کدی که قراره با قوانین Emmet بزنیم خیلی طولانی میشه و ممکنه گمراهمون کنه و وسط کد زدن متوجه نشیم الان دقیقا در کجای کار هستیم. میتوانیم با استفاده از خاصیت grouping کد ها رو دسته بندی کنیم.

برای استفاده از این ویژگی باید از () استفاده کنیم.

این ویژگی باعث میشه مرتب تر کد بزنیم و کمک میکنه کمتر از ^

```
index.html > ...
<section>
  <div class="picture-container" id="pic-1">
    <img src="" alt="">
  </div>

  <div class="picture-container" id="pic-2">
    <img src="" alt="">
  </div>
</section>

<section>
  <ul>
    <li></li>
    <li></li>
    <li></li>
    <li></li>
  </ul>
</section>
```

استفاده کنیم.

```
(section>div.picture-container#pic-1>img^div.picture-container#pic-2>img)+(section>ul>li*4)
```



```
<section>
  <div class="picture-container" id="pic-1"><img src="" alt=""></div>
  <div class="picture-container" id="pic-2"><img src="" alt=""></div>
</section>
<section>
  <ul>
    <li></li>
    <li></li>
    <li></li>
    <li></li>
  </ul>
</section>
```

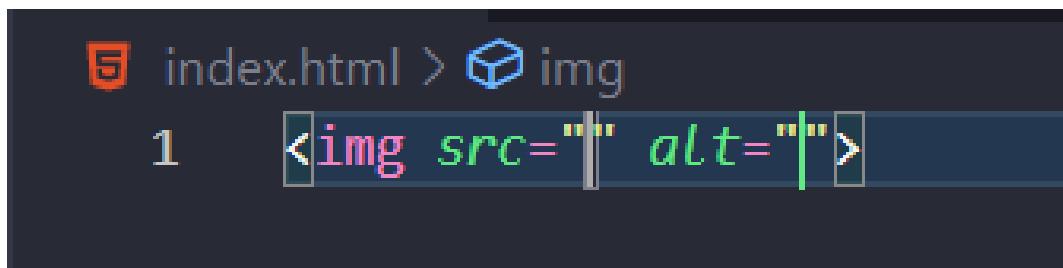
دقت کنید وقتی خواستم دو تا تگ section هم سطح هم باشند،

دیگه از ^ استفاده نکردم و خاصیت Grouping این امکان رو داد

تا هم سطح قرار بگیرند.

## Attribute – 11

برخی از اتریبیوت ها را IDE یا Text editor به صورت خودکار برآتون میسازه. مثلا تگ img را فرض کنیم :



A screenshot of a code editor window titled "index.html > img". The code editor shows the following line of HTML:

```
1 <img src="" alt="">
```

The "src" and "alt" attributes are highlighted in green, indicating they were generated automatically by the IDE.

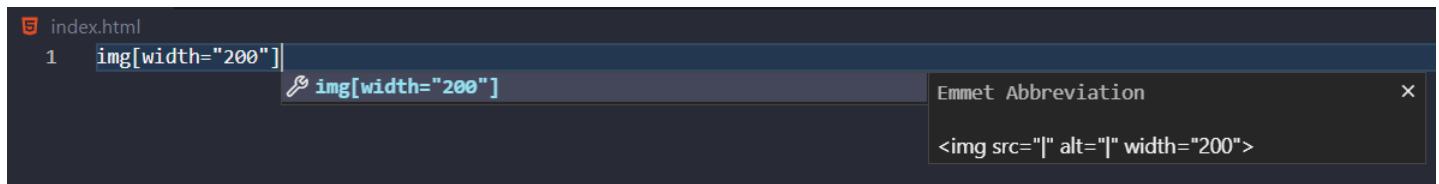
میبینید که اتریبیوت های src و alt را به طور اتوماتیک نوشته شده.

اما مثلا شما به اتریبیوت width هم نیاز دارید.

برای اضافه کردن اتریبیوت باید از فرمول زیر استفاده کنید :

TagName[attribute]

## مثال



index.html

```
1  img[width="200"]
```

Emmet Abbreviation

```

```



index.html > ...

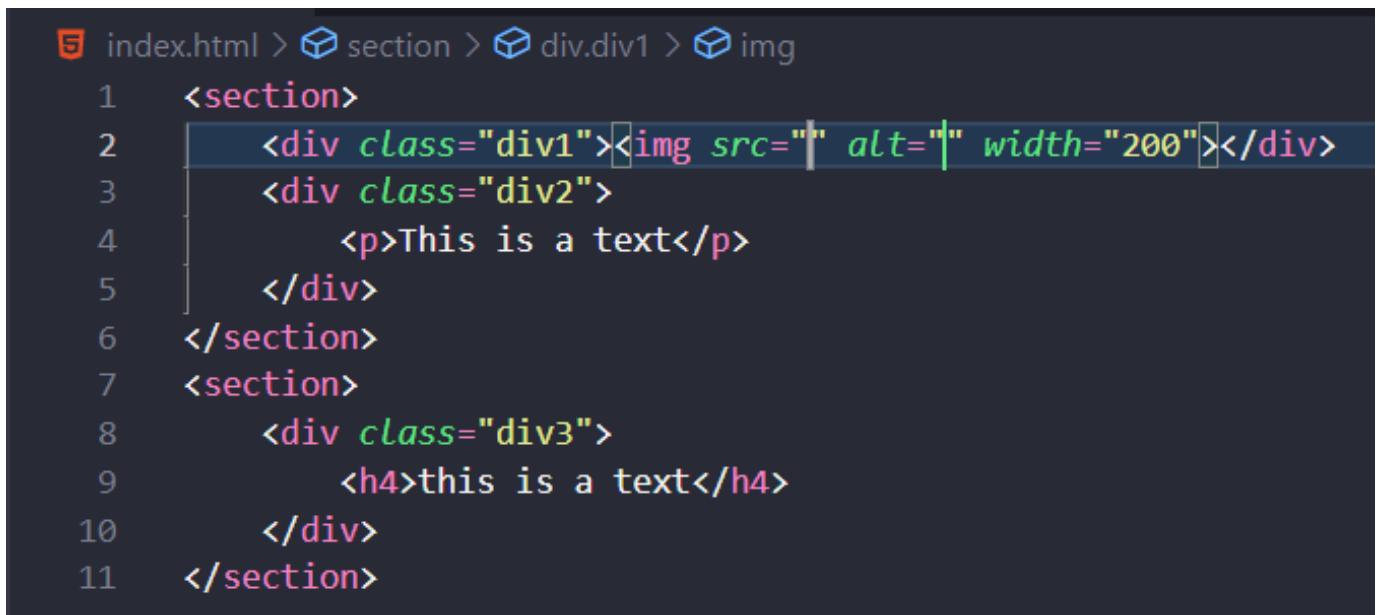
```
1  <img src="" alt="" width="200">
2
```

نتیجه

یک مثال ترکیبی از هر چه آموختیم :

```
(section>div.div1>img[width="200"]^div.div2>p{This is a text})+  
(section>div.div3>h4{this is a text})
```

به نظرتون خروجی کد بالا چیه ؟



index.html > section > div.div1 > img

```
1  <section>
2    <div class="div1"></div>
3    <div class="div2">
4      <p>This is a text</p>
5    </div>
6  </section>
7  <section>
8    <div class="div3">
9      <h4>this is a text</h4>
10     </div>
11   </section>
```

## Numbering – 12

تکه کد پایین رو ببینید :

```
5 index.html > ...
1   <ul>
2     <li class="item-1"></li>
3     <li class="item-2"></li>
4     <li class="item-3"></li>
5     <li class="item-4"></li>
6   </ul>
```

به اسم کلاس ها دقت کنید که به ترتیب هستند.

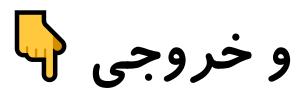
ممکنه بعضی وقت ها تعداد کلاس ها زیاد باشه و شماره ها بیشتر بشه و خب امکان خطأ هم بالاتر میره و یا کلا خیلی وقت گیره که شما یکی یکی به تگ هاتون کلاس بدید.

عزیز کار رو برای ما ساده کرده و میتوانید از ⚡ برای شماره گذاری استفاده کنید.

تکه کد بالا رو با استفاده از قوانین Emmet میتوانیم به شکل زیر بنویسیم



```
ul>li.item-$*4
```



```
<ul>
  <li class="item-1"></li>
  <li class="item-2"></li>
  <li class="item-3"></li>
  <li class="item-4"></li>
</ul>
```

راستی میدونید اگه بجای \$- خود \$ رو استفاده میکردیم، خروجی

چی میشد؟



خودتون تست کنید

حتی شما این امکان رو دارید که در text ها از این ویژگی استفاده کنید.



```
ul>li.item-$*4{ this is list number $ }
```

همونطور که میبینید در قسمت متن فقط از علامت \$ استفاده کردم.



```
<ul>
  <li class="item-1"> this is list number 1 </li>
  <li class="item-2"> this is list number 2 </li>
  <li class="item-3"> this is list number 3 </li>
  <li class="item-4"> this is list number 4 </li>
</ul>
```



واقعا خیلی خیلی کار رو راحت کرده

مثال

```
index.html
1  p{This is text number $}*5
2
```

نتیجه

```
index.html > p
1  <p>This is text number 1</p>
2  <p>This is text number 2</p>
3  <p>This is text number 3</p>
4  <p>This is text number 4</p>
5  <p>This is text number 5</p>
6
```

مثال

```
index.html
1  p{This is text number $$$$$}*5
2
3
```

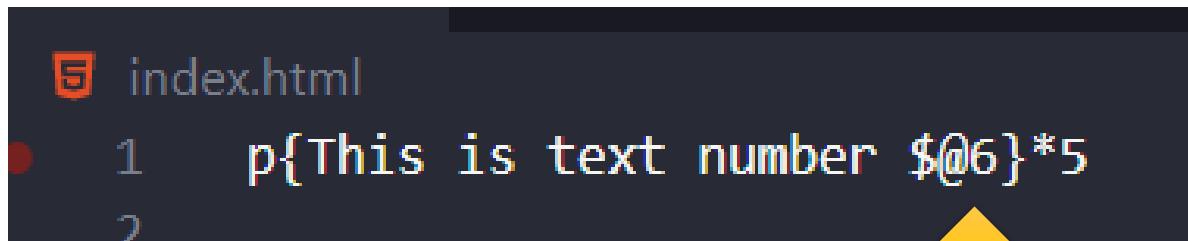
توجه کنید که به جای یک علامت دلار، از 4 علامت استفاده کردم.

نتیجه

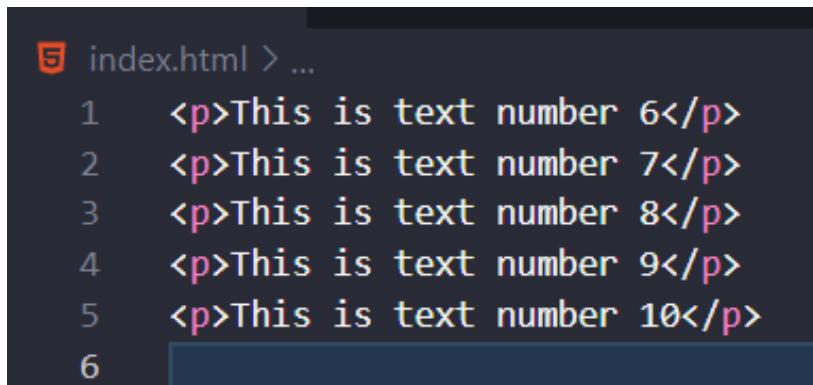
```
index.html > p
1  <p>This is text number 0001</p>
2  <p>This is text number 0002</p>
3  <p>This is text number 0003</p>
4  <p>This is text number 0004</p>
5  <p>This is text number 0005</p>
6
7
8
```

بعضی وقت ها شاید ما بخواهیم شمارش و نام گذاری از یک عدد خاص شروع بشه.

به مثال زیر دقت کنید



```
index.html
1 p{This is text number ${@6}*5
2
```



```
index.html > ...
1 <p>This is text number 6</p>
2 <p>This is text number 7</p>
3 <p>This is text number 8</p>
4 <p>This is text number 9</p>
5 <p>This is text number 10</p>
6
```

نتیجه

همونطور که میبینید از شماره 6 شروع کرد.

تگ های اینپوت خیلی خیلی در فرم ها پر استفاده هستند.

این تگ ها type های زیادی دارند که بعضی وقت ها ممکنه فراموش بشه.

کار ما رو خیلی راحت کرده. شما با استفاده از دستور زیر میتوانید تگ های اینپوت رو خیلی راحت ازشون استفاده کنید :

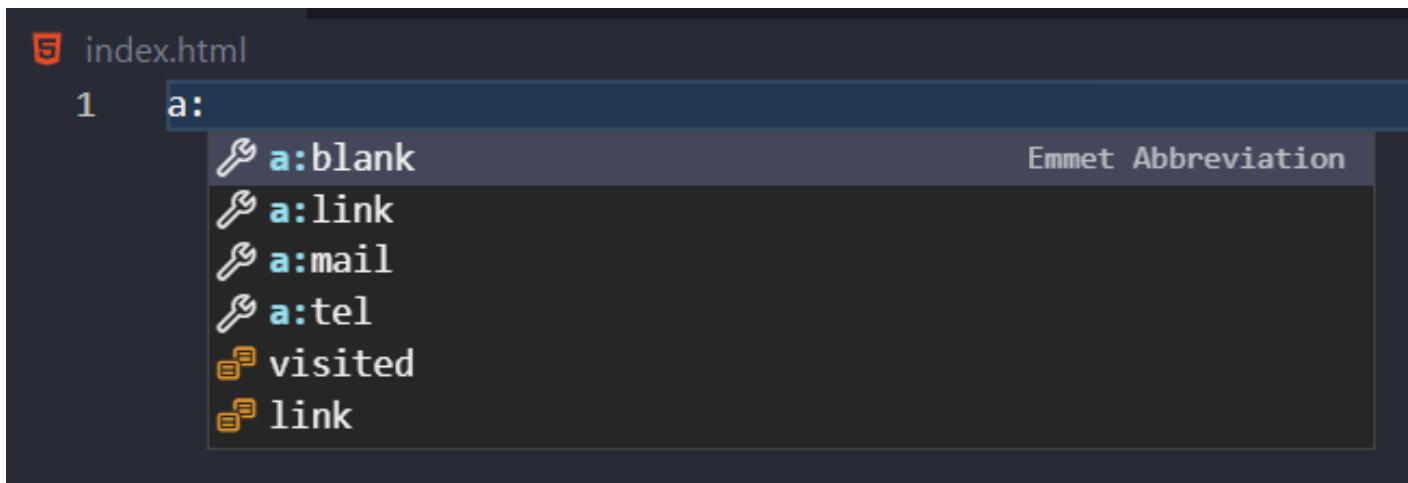
### Input:type



به عکس زیر دقت کنید

The screenshot shows a code editor window with the file 'index.html' open. The code '1 input:' is typed. A dropdown menu titled 'Emmet Abbreviation' lists various input types: 'input:b', 'input:btn', 'input:button', 'input:c', 'input:checkbox', 'input:color', 'input:date', 'input:datetime', 'input:datetime-local', 'input:email', 'input:f', and 'input:file'. The suggestion 'input:button' is highlighted. To the right of the dropdown, the resulting Emmet abbreviation is shown: '<input type="button" value="">'.

دقیقا از همین روش برای تگ هم میتوانید استفاده کنید :



The screenshot shows a code editor window with the file name 'index.html'. In the code area, there is a single line of code: '1 a:'. A dropdown menu titled 'Emmet Abbreviation' is open, listing several options:

- a:blank
- a:link
- a:mail
- a:tel
- visited
- link

Link – 14

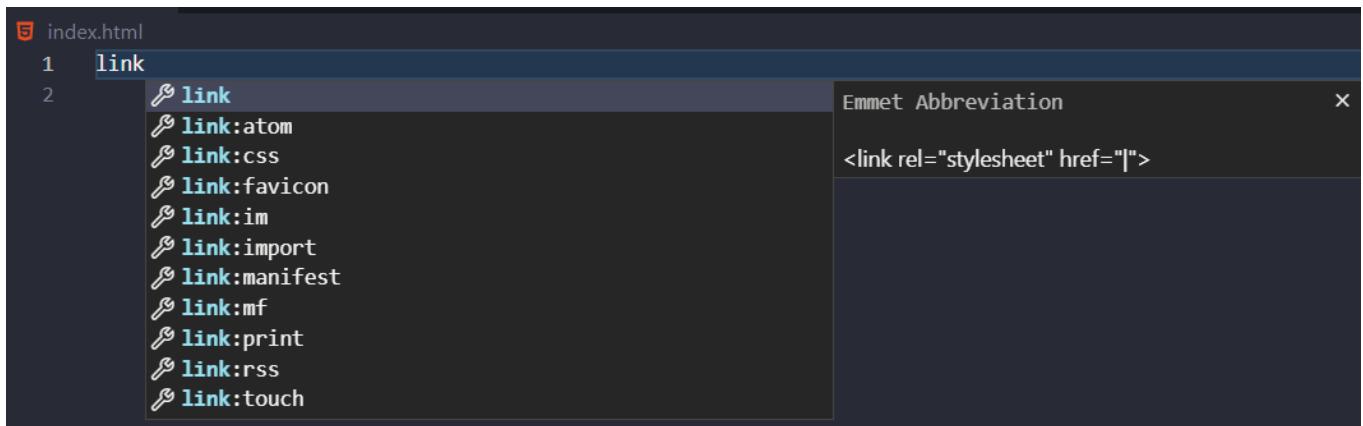
در طراحی وب سایت، از لینک ها خیلی زیاد استفاده میشه.

کار با لینک ها رو خیلی برای ما ساده کرده.

فرمول استفاده از لینک ها به شکل زیر است :

Link: ....

به عکس زیر دقت کنید



همونطور که میبینید انواع لینک ها رو شما به راحتی میتوانید استفاده کنید و دیگه نیازی نیست تا یک تگ رو به طور کامل و از صفر بنویسید.

# CS5



## Margin, Padding – 15

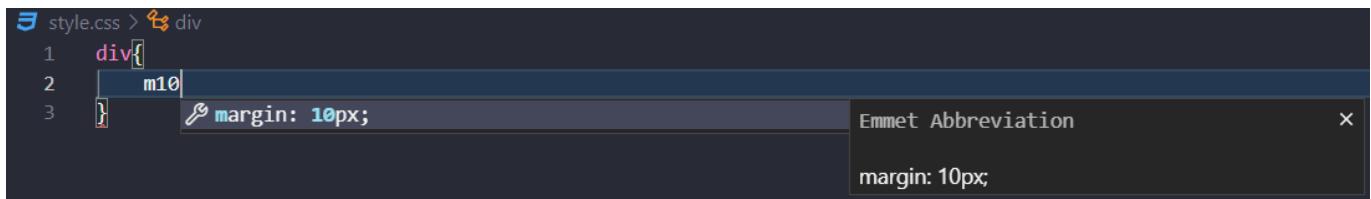
برای استفاده‌ی خیلی راحت تر از padding و margin در فایل‌های

CSS میتوانید از فرمول پایین استفاده کنید :

M(value) ↳ margin: value ;

P(value) ↳ padding: value ;

مثال



```
style.css > div
1  div[
2    m10|
3  ]
```

Emmet Abbreviation

margin: 10px;



در کل داریم :

```
m : margin
mt : margin-top
mb : margin-bottom
ml : margin-left
mr : margin-right
-----
p : padding
pt : padding-top
pb : padding-bottom
pl : padding-left
pr : padding-right
```

حتماً تا به حال از padding یا margin به شکل زیر استفاده کردید:

```
style.css > div
1 < div{
2   margin: 10px 20px 30px 40px;
3   padding: 10px 20px 30px 40px;
4 }
```

کدهای بالا، طبق قوانین Emmet

```
m10px20px30px40px
}
margin: 10px 20px 30px 40px; Emmet Abbreviation
```

```
p10px20px30px40px
}
padding: 10px 20px 30px 40px; Emmet Abbreviation
```

همونطور که همه‌ی توسعه دهندگان فراتر اطلاع دارند، CSS خیلی خیلی گسترده است و اصلاً نمی‌شود تمام مطالبش را بیان کرد.

در ارتباط با قوانین Emmet در CSS، باید بیان کنم که قوانین بسیار زیادی وجود داره که حفظ کردن اونها کار درستی نیست.

در انتهای این فایل Cheat Sheet خیلی کاملی از قوانین Emmet برآتون قرار دادم که می‌توانید به اون مراجعه کنید تا بیشتر با قوانین این پلاگین در رابطه با CSS آشنا شوید.

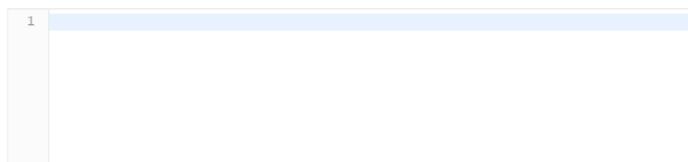
اگر داکیومنت اصلی Emmet را مطالعه کنید، به قسمت خیلی جالبی میرسید به نام Fuzzy search؛ در این قسمت شما می‌توانید اختصار های CSS را بنویسید و برآتون کاملش می‌کنید.

### Fuzzy search

If you take a look at [Cheat Sheet](#), you'll find that there are too many CSS snippets to remember. Also, some of them might be a bit lengthy for sake of logical separation.

To make CSS writing a bit easier, Emmet implement *fuzzy search* logic for CSS snippets: every time you enter unknown abbreviation, Emmet will try to find a closest snippet definition.

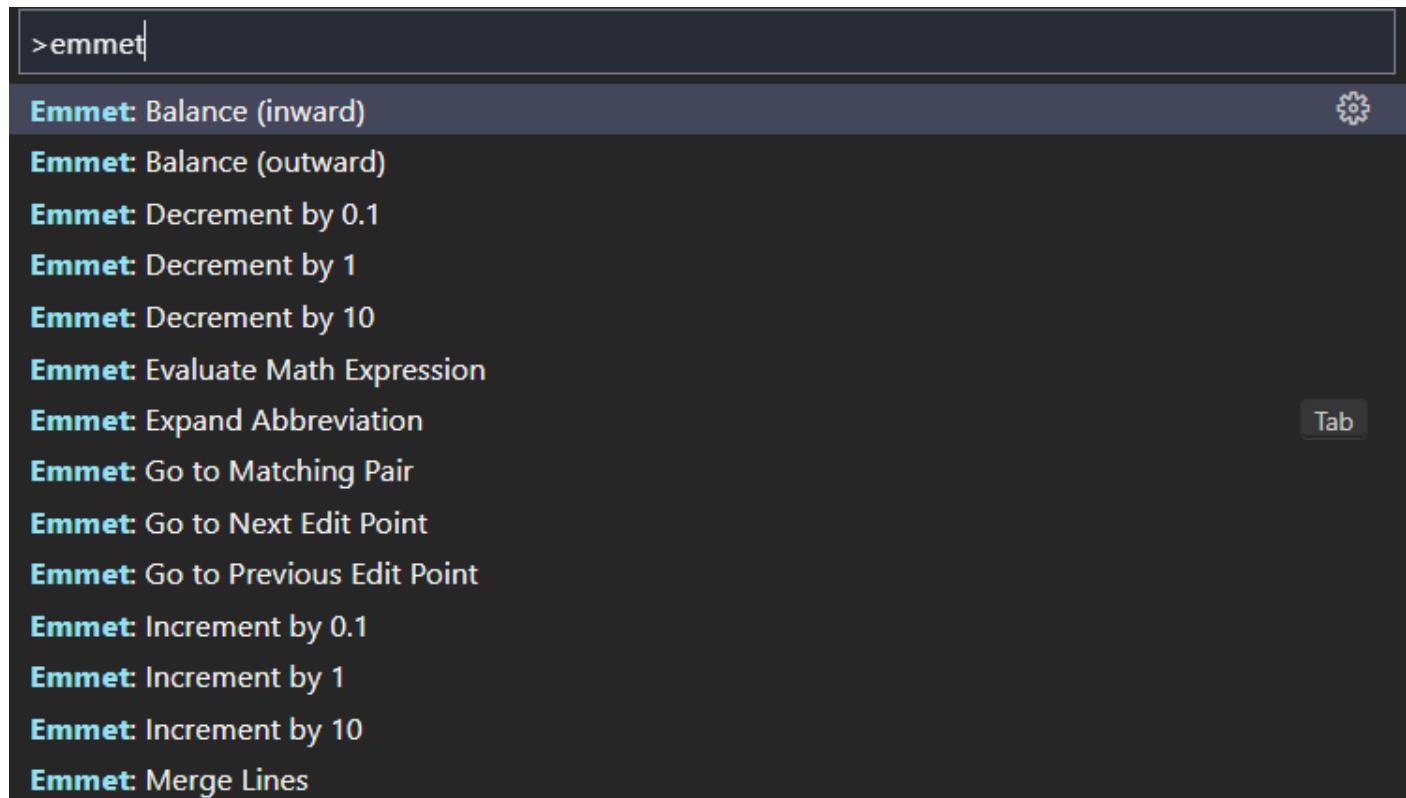
For example, instead of writing `ov:h (overflow: hidden;)` abbreviation, you can write `ov-h`, `ovh` or even `oh`. You can play around with the fuzzy search in text editor below. Try to find as many variations as possible (use Tab key to expand abbreviations) for `bxz:cb`, `ovx:h` and `pos:a` snippets.



The fuzzy search is performed against *predefined snippet names*, not snippet values or CSS properties. This results in more predictable and controllable matches. Remember that you can always [create your own snippets](#) or [redefine existing ones](#) to fine-tune fuzzy search experience.

من همیشه در فایل هایی که مینویسم، قسمتی از یادگیری رو به خواننده‌ی فایل میسپارم چون اعتقاد دارم خواندن یک مطلب باعث یادگیری نخواهد شد و تمرین کردن عامل یادگیری است.

اگر command pallet رو باز کنید و در اون Emmet رو بنویسید، با گزینه‌های خیلی جالبی روبرو خواهید شد. به عکس زیر دقت کنید.



این گزینه ها، کار هایی هستند که Emmet میتوانه انجام بده و تا به حال اطلاعی از شون نداشتیم.

حتماً به شون سر بزنید و امتحانشون کنید.

برای اینکه راحت تر یاد بگیرید، در داکیومنت Emmet به قسمت Action برد و اونجا با مثال میتوانید یاد بگیرید.

#### Actions

- Expand Abbreviation
- Balance
- Go to Matching Pair
- Wrap with Abbreviation
- Go to Edit Point
- Select Item
- Toggle Comment
- Split/Join Tag
- Remove Tag
- Merge Lines
- Update Image Size
- Evaluate Math Expression
- Increment/Decrement Number
- Reflect CSS Value
- Encode/Decode Image to data:URL

در ضمن باید این نکته رو بیان کنم که بیشترین استفاده از Emmet در نوشتن کدهای HTML است و در CSS کمتر استفاده میشود.

## 16 – منابع آموزشی

- ✓ بهترین منبع یادگیری : وب سایت رسمی Emmet و
- دکیومنت اصلی
- ✓ وبسایت WebProg
- ✓ یوتوب

# Emmet

# Cheat Sheet

**SYNTAX**

<b>Syntax</b>	<b>1</b>
Child: >	1
Sibling: +	1
Climb-up: ^	1
Grouping: ()	1
Multiplication: *	2
Naming and numbering: \$	2
ID and CLASS attributes	2
Custom attributes	2
Text: {}	2
Implicit tag names	2
<b>HTML</b>	<b>3</b>
Form, Input, Button	4
<b>CSS</b>	<b>7</b>
Visual Formatting	7
Margin & Padding	9
Box Sizing	9
Font	9
Text	11
Background	13
Color	14
Generated content	14
Outline	15
Tables	15
Border	15
Lists	18
Print	18
Others	18
Animation	19
Flex	20
Transform, Transition	21
<b>XSL</b>	<b>22</b>
<b>HTML DOCTYPES</b>	<b>23</b>

**Child: >****nav>ul>li**

```
<nav>
  <ul>
    <li></li>
  </ul>
</nav>
```

**Sibling: +****div+p+bq**

```
<div></div>
<p></p>
<blockquote></blockquote>
```

**Climb-up: ^****div+div>p>span+em^bq**

```
<div></div>
<div>
  <p><span></span><em></em></p>
  <blockquote></blockquote>
</div>
```

**div+div>p>span+em^bq**

```
<div></div>
<div>
  <p><span></span><em></em></p>
</div>
<blockquote></blockquote>
```

**Grouping: ()****div>(header>ul>li\*2>a)+footer>p**

```
<div>
  <header>
    <ul>
      <li><a href=""></a></li>
      <li><a href=""></a></li>
    </ul>
  </header>
  <footer>
    <p></p>
  </footer>
</div>
```

**(div>dl>(dt+dd)\*3)+footer>p**

```
<div>
  <dl>
    <dt></dt>
    <dd></dd>
    <dt></dt>
    <dd></dd>
    <dt></dt>
    <dd></dd>
  </dl>
</div>
```

```
<footer>
  <p></p>
</footer>
```

**Multiplication: \***

```
ul>li*5
<ul>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

**Naming and numbering: \$**

```
ul>li.sample$*5
```

```
<ul>
  <li class="sample1"></li>
  <li class="sample2"></li>
  <li class="sample3"></li>
  <li class="sample4"></li>
  <li class="sample5"></li>
</ul>
```

```
h${[title=topic${}]{Headline ${}}*3
```

```
<h1 title="topic1">Headline 1</h1>
<h2 title="topic2">Headline 2</h2>
<h3 title="topic3">Headline 3</h3>
```

```
ul>li.item${}*5
```

```
<ul>
  <li class="item001"></li>
  <li class="item002"></li>
  <li class="item003"></li>
  <li class="item004"></li>
  <li class="item005"></li>
</ul>
```

```
ul>li.item${@}*5
```

```
<ul>
  <li class="item5"></li>
  <li class="item4"></li>
  <li class="item3"></li>
  <li class="item2"></li>
  <li class="item1"></li>
</ul>
```

```
ul>li.item${@3}*5
```

```
<ul>
  <li class="item3"></li>
  <li class="item4"></li>
  <li class="item5"></li>
  <li class="item6"></li>
  <li class="item7"></li>
</ul>
```

**ID and CLASS attributes**

```
#header
```

```
<div id="header"></div>
```

```
.title
```

```
<div class="title"></div>
```

```
form#search.wide
```

```
<form id="search" class="wide"></form>
```

```
p.class1.class2.class3
```

```
<p class="class1 class2 class3"></p>
```

**Custom attributes**

```
p[title="Hello world"]
```

```
<p title="Hello world"></p>
```

```
td[rowspan=2 colspan=3 title]
```

```
<td rowspan="2" colspan="3" title=""></td>
```

```
[a='value1' b="value2"]
```

```
<div a="value1" b="value2"></div>
```

**Text: {}**

```
a{Click me}
```

```
<a href="">Click me</a>
```

```
p>{Click }+a{here}+{ to continue}
```

```
<p>Click <a href="">here</a> to continue</p>
```

**Implicit tag names**

```
.class
```

```
<div class="class"></div>
```

```
em>.class
```

```
<em><span class="class"></span></em>
```

```
ul>.class
```

```
<ul>
  <li class="class"></li>
</ul>
```

```
table>.row>.col
```

```
<table>
  <tr class="row">
    <td class="col"></td>
  </tr>
</table>
```

# HTML

All unknown abbreviations will be transformed to tag, e.g. foo → <foo></foo>.

**a**  
  <a href=""></a>

**a:link**  
  <a href="http://"></a>

**a:mail**  
  <a href="mailto:"></a>

**abbr**  
  <abbr title=""></abbr>

**acronym, acr**  
  <acronym title=""></acronym>

**base**  
  <base href="" />

**basefont**  
  <basefont />

**br**  
  <br />

**frame**  
  <frame />

**hr**  
  <hr />

**bdo**  
  <bdo dir=""></bdo>

**bdo:r**  
  <bdo dir="rtl"></bdo>

**bdo:l**  
  <bdo dir="ltr"></bdo>

**col**  
  <col />

**link**  
  <link rel="stylesheet" href="" />

**link:css**  
  <link rel="stylesheet" href="style.css" />

**link:print**  
  <link rel="stylesheet" href="print.css" media="print" />

**link:favicon**  
  <link rel="shortcut icon" type="image/x-icon" href="favicon.ico" />

**link:touch**  
  <link rel="apple-touch-icon" href="favicon.png" />

3

**link:rss**

```
<link rel="alternate"
      type="application/rss+xml"
      title="RSS" href="rss.xml" />
```

**link:atom**

```
<link rel="alternate"
      type="application/atom+xml"
      title="Atom" href="atom.xml" />
```

**link:import, link:im**

```
<link rel="import"
      href="component.html" />
```

**meta**

```
<meta />
```

**meta:utf**

```
<meta http-equiv="Content-Type"
      content="text/html; charset=UTF-8"
      />
```

**meta:win**

```
<meta http-equiv="Content-Type"
      content="text/html; charset=windows-1251" />
```

**meta:vp**

```
<meta name="viewport"
      content="width=device-width, user-scalable=no, initial-scale=1.0,
      maximum-scale=1.0, minimum-scale=1.0" />
```

**meta:compat**

```
<meta http-equiv="X-UA-Compatible"
      content="IE=7" />
```

**style**

```
<style></style>
```

**script**

```
<script></script>
```

**script:src**

```
<script src=""></script>
```

**img**

```
<img src="" alt="" />
```

**img:srcset, img:s**

```
<img srcset="" src="" alt="" />
```

**img:sizes, img:z**

```
<img sizes="" srcset="" src=""
      alt="" />
```

**picture**

```
<picture></picture>
```

**source, src**

```
<source />
```

**source:src, src:sc**

```
<source src="" type="" />
```

```

source:srcset, src:s
  <source srcset="" />
source:media, src:m
  <source media="(min-width: )"
  srcset="" />
source:type, src:t
  <source srcset="" type="image/" />
source:sizes, src:z
  <source sizes="" srcset="" />
source:media:type, src:mt
  <source media="(min-width: )"
  srcset="" type="image/" />
source:media:sizes, src:mz
  <source media="(min-width: )"
  sizes="" srcset="" />
source:sizes:type, src:zt
  <source sizes="" srcset=""
  type="image/" />
iframe
  <iframe src="" frameborder="0"></
  iframe>
embed
  <embed src="" type="" />
object
  <object data="" type=""></object>
param
  <param name="" value="" />
map
  <map name=""></map>
area
  <area shape="" coords="" href="" alt="" />
area:d
  <area shape="default" href="" alt="" />
area:c
  <area shape="circle" coords="" href="" alt="" />
area:r
  <area shape="rect" coords="" href="" alt="" />
area:p
  <area shape="poly" coords="" href="" alt="" />

```

**Form, Input, Button**

**form**  
 <form action=""></form>

**form:get**  
 <form action="" method="get"></
 form>

**form:post**  
 <form action="" method="post"></
 form>

**label**  
 <label for=""></label>

**input**  
<input type="text" />

**inp**  
<input type="text" name="" id="" />

**input:hidden, input:h**  
<input type="hidden" name="" />

**input:text, input:t**  
<input type="text" name="" id="" />

**input:search**  
<input type="search" name="" id="" />

**input:email**  
<input type="email" name="" id="" />

**input:url**  
<input type="url" name="" id="" />

**input:password, input:p**  
<input type="password" name="" id="" />

**input:datetime**  
<input type="datetime" name="" id="" />

**input:date**  
<input type="date" name="" id="" />

**input:datetime-local**  
<input type="datetime-local" name="" id="" />

**input:month**  
<input type="month" name="" id="" />

**input:week**  
<input type="week" name="" id="" />

**input:time**  
<input type="time" name="" id="" />

**input:tel**      <input type="tel" name="" id="" />

**input:number**    <input type="number" name="" id="" />

**input:color**     <input type="color" name="" id="" />

**input:checkbox, input:c**    <input type="checkbox" name="" id="" />

**input:radio, input:r**    <input type="radio" name="" id="" />

**input:range**     <input type="range" name="" id="" />

**input:file, input:f**    <input type="file" name="" id="" />

**input:submit, input:s**    <input type="submit" value="" />

**input:image, input:i**    <input type="image" src="" alt="" />

**input:button, input:b**    <input type="button" value="" />

**isindex**        <isindex />

**input:reset**     <input type="reset" value="" />

**select**        <select name="" id="" />

**select:disabled, select:d**    <select name="" id="" disabled="disabled" />

**option, opt**     <option value="" />

**textarea**       <textarea name="" id="" cols="30" rows="10" />

**marquee**       <marquee behavior="" direction="" />

**menu:context, menu:c**    <menu type="context" />

**menu:toolbar, menu:t**    <menu type="toolbar" />

**video**        <video src="" />

**audio**        <audio src="" />

**html:xml**       <html xmlns="http://www.w3.org/1999/xhtml" />

**keygen**       <keygen />

**command**      <command />

**button:submit, button:s, btn:s**    <button type="submit" />

**button:reset, button:r, btn:r**    <button type="reset" />

**button:disabled, button:d, btn:d**    <button disabled="disabled" />

**fieldset:disabled, fieldset:d, fset:d**    <fieldset disabled="disabled" />

**bq**        <blockquote />

**fig**        <figure />

**figcaption**    <figcaption />

**pic**        <picture />

**ifr**        <iframe src="" frameborder="0" />

**emb**        <embed src="" type="" />

**obj**        <object data="" type="" />

**cap**        <caption />

**colg**       <colgroup />

**fieldset, fset**    <fieldset />

**btn**        <button />

**optgroup**    <optgroup />

**tarea**       <textarea name="" id="" cols="30" rows="10" />

```

leg <legend></legend>
sect <section></section>
art <article></article>
hdr <header></header>
ftr <footer></footer>
adr <address></address>
dlg <dialog></dialog>
str <strong></strong>
prog <progress></progress>
mn <main></main>
tem <template></template>
datag <datagrid></datagrid>
datal <datalist></datalist>
kg <keygen />
out <output></output>
det <details></details>
cmd <command />
ri:dpr, ri:d <img srcset="" src="" alt="" />
ri:viewport, ri:v <img sizes="" srcset="" src="" alt="" />
ri:art, ri:a <picture>
    <source media="(min-width: )" srcset="" />
    <img src="" alt="" />
</picture>

```

```

ri:type, ri:t <picture>
    <source srcset="" type="image/" />
    <img src="" alt="" />
</picture>
ol+ <ol>
    <li></li>
</ol>
ul+ <ul>
    <li></li>
</ul>
dl+ <dl>
    <dt></dt>
    <dd></dd>
</dl>
map+ <map name="">
    <area shape="" coords="" href="" alt="" />
</map>
table+ <table>
    <tr>
        <td></td>
    </tr>
</table>
colgroup+, col+ <colgroup>
    <col />
</colgroup>
tr+ <tr>
    <td></td>
</tr>
select+ <select name="" id="">
    <option value=""></option>
</select>
optgroup+, optg+ <optgroup>
    <option value=""></option>
</optgroup>
pic+ <picture>
    <source srcset="" />
    <img src="" alt="" />
</picture>

```

## CSS

CSS module uses fuzzy search to find unknown abbreviations, e.g. ov:h == ov-h == ovh == oh.

If abbreviation wasn't found, it is transformed into property name: foo-bar → foo-bar: l;

You can prefix abbreviations with hyphen to produce vendor-prefixed properties: -foo

### Visual Formatting

**pos** position:relative;  
**pos:s** position:static;  
**pos:a** position:absolute;  
**pos:r** position:relative;  
**pos:f** position:fixed;  
**t** top:;  
**t:a** top:auto;  
**r** right:;  
**r:a** right:auto;  
**b** bottom:;  
**b:a** bottom:auto;  
**l** left:;  
**l:a** left:auto;  
**z** z-index:;  
**z:a** z-index:auto;  
**fl** float:left;  
**fl:n** float:none;  
**fl:l** float:left;

**fl:r** float:right;  
**cl** clear:both;  
**cl:n** clear:none;  
**cl:l** clear:left;  
**cl:r** clear:right;  
**cl:b** clear:both;  
**d** display:block;  
**d:n** display:none;  
**d:b** display:block;  
**d:f** display:flex;  
**d:if** display:inline-flex;  
**d:i** display:inline;  
**d:ib** display:inline-block;  
**d:li** display:list-item;  
**d:ri** display:run-in;  
**d:cp** display:compact;  
**d:tb** display:table;  
**d:itb** display:inline-table;  
**d:tbcp** display:table-caption;  
**d:tbcl** display:table-column;  
**d:tbclg** display:table-column-group;  
**d:tbhg** display:table-header-group;

**d:tbfg** display:table-footer-group;  
**d:tbr** display:table-row;  
**d:tbrg** display:table-row-group;  
**d:tbc** display:table-cell;  
**d:rb** display:ruby;  
**d:rbb** display:ruby-base;  
**d:rbbg** display:ruby-base-group;  
**d:rbt** display:ruby-text;  
**d:rbtg** display:ruby-text-group;

**v** visibility:hidden;  
**v:v** visibility:visible;  
**v:h** visibility:hidden;  
**v:c** visibility:collapse;  
**ov** overflow:hidden;  
**ov:v** overflow:visible;  
**ov:h** overflow:hidden;  
**ov:s** overflow:scroll;  
**ov:a** overflow:auto;  
**ovx** overflow-x:hidden;  
**ovx:v** overflow-x:visible;  
**ovx:h** overflow-x:hidden;  
**ovx:s** overflow-x:scroll;

**ovx:a** overflow-x:auto;  
**ovy** overflow-y:hidden;  
**ovy:v** overflow-y:visible;  
**ovy:h** overflow-y:hidden;  
**ovy:s** overflow-y:scroll;  
**ovy:a** overflow-y:auto;  
**ovs** overflow-style:scrollbar;  
**ovs:a** overflow-style:auto;  
**ovs:s** overflow-style:scrollbar;  
**ovs:p** overflow-style:panner;  
**ovs:m** overflow-style:move;  
**ovs:mq** overflow-style:marquee;  
**zoo, zm** zoom:1;  
**cp** clip:;  
**cp:a** clip:auto;  
**cp:r** clip:rect(top right bottom left);  
**rsz** resize:;  
**rsz:n** resize:none;  
**rsz:b** resize:both;  
**rsz:h** resize:horizontal;  
**rsz:v** resize:vertical;  
**cur** cursor:\${pointer};

```

cur:a    cursor:auto;
cur:d    cursor:default;
cur:c    cursor:crosshair;
cur:ha   cursor:hand;
cur:he   cursor:help;
cur:m    cursor:move;
cur:p    cursor:pointer;
cur:t    cursor:text;

```

## Margin & Padding

```

m        margin:;;
m:a     margin:auto;
mt      margin-top:;;
mt:a   margin-top:auto;
mr      margin-right:;;
mr:a   margin-right:auto;
mb      margin-bottom:;;
mb:a   margin-bottom:auto;
ml      margin-left:;;
ml:a   margin-left:auto;
p        padding:;;
pt     padding-top:;;
pr     padding-right:;;
pb     padding-bottom:;;

```

```

pl      padding-left:;;

```

## Box Sizing

```

bz      box-sizing:border-box;
bz:cb   box-sizing:content-box;
bz:bb   box-sizing:border-box;
bxsh   box-shadow:inset hoff voff blur
          color;
bxsh:r  box-shadow:inset hoff voff blur
          spread rgb(0, 0, 0);
bxsh:ra box-shadow:inset h v blur spread
          rgba(0, 0, 0, .5);
bxsh:n  box-shadow:none;
w        width:;;
w:a     width:auto;
h        height:;;
h:a     height:auto;
maw    max-width:;;
maw:n  max-width:none;
mah    max-height:;;
mah:n  max-height:none;
miw   min-width:;;
mih   min-height:;;

```

## Font

```

f        font:;;

```

**f+** font:1em Arial,sans-serif;

**fw** font-weight:;

**fw:n** font-weight:normal;

**fw:b** font-weight:bold;

**fw:br** font-weight:bolder;

**fw:lr** font-weight:lighter;

**fs, fs:i** font-style:italic;

**fs:n** font-style:normal;

**fs:o** font-style:oblique;

**fv** font-variant:;

**fv:n** font-variant:normal;

**fv:sc** font-variant:small-caps;

**fz** font-size:;

**fza** font-size-adjust:;

**fza:n** font-size-adjust:none;

**ff** font-family:;

**ff:s** font-family:serif;

**ff:ss** font-family:sans-serif;

**ff:c** font-family:cursive;

**ff:f** font-family:fantasy;

**ff:m** font-family:monospace;

**ff:a** font-family: Arial, „Helvetica Neue“, Helvetica, sans-serif;

**ff:t** font-family: „Times New Roman“, Times, Baskerville, Georgia, serif;

**ff:v** font-family: Verdana, Geneva, sans-serif;

**fef** font-effect:;

**fef:n** font-effect:none;

**fef:eg** font-effect:engrave;

**fef:eb** font-effect:emboss;

**fef:o** font-effect:outline;

**fem** font-emphasize:;

**femp** font-emphasize-position:;

**femp:b** font-emphasize-position:before;

**femp:a** font-emphasize-position:after;

**fems** font-emphasize-style:;

**fems:n** font-emphasize-style:none;

**fems:ac** font-emphasize-style:accent;

**fems:dt** font-emphasize-style:dot;

**fems:c** font-emphasize-style:circle;

**fems:ds** font-emphasize-style:disc;

**fsm** font-smooth:;

**fsm:a** font-smooth:auto;

**fsm:n** font-smooth:never;

**fsm:aw** font-smooth:always;

```

fst          font-stretch:;
fst:n        font-stretch:normal;
fst:uc       font-stretch:ultra-condensed;
fst:ec       font-stretch:extra-condensed;
fst:c        font-stretch:condensed;
fst:sc       font-stretch:semi-condensed;
fst:se       font-stretch:semi-expanded;
fst:e        font-stretch:expanded;
fst:ee       font-stretch:extra-expanded;
fst:ue       font-stretch:ultra-expanded;

```

## Text

```

va          vertical-align:top;
va:sup      vertical-align:super;
va:t        vertical-align:top;
va:tt       vertical-align:text-top;
va:m        vertical-align:middle;
va:bl       vertical-align:baseline;
va:b        vertical-align:bottom;
va:tb       vertical-align:text-bottom;
va:sub      vertical-align:sub;
ta, ta:l    text-align:left;
ta:c        text-align:center;

```

```

ta:r         text-align:right;
ta:j         text-align:justify;
ta-lst      text-align-last:;
tal:a       text-align-last:auto;
tal:l        text-align-last:left;
tal:c        text-align-last:center;
tal:r        text-align-last:right;
td, td:n    text-decoration:none;
td:u        text-decoration:underline;
td:o        text-decoration:overline;
td:l        text-decoration:line-through;
te          text-emphasis:;
te:n        text-emphasis:none;
te:ac       text-emphasis:accent;
te:dt       text-emphasis:dot;
te:c        text-emphasis:circle;
te:ds       text-emphasis:disc;
te:b        text-emphasis:before;
te:a        text-emphasis:after;
th          text-height:;
th:a        text-height:auto;
th:f        text-height:font-size;

```

```

th:t      text-height:text-size;
th:m      text-height:max-size;
ti        text-indent:;
ti:-     text-indent:-9999px;
tj        text-justify:;
tj:a     text-justify:auto;
tj:iw    text-justify:inter-word;
tj:ii    text-justify:inter-ideograph;
tj:ic    text-justify:inter-cluster;
tj:d     text-justify:distribute;
tj:k     text-justify:kashida;
tj:t     text-justify:tibetan;
to       text-outline:;
to+     text-outline:0 0 #000;
to:n     text-outline:none;
tr       text-replace:;
tr:n    text-replace:none;
tt       text-transform:uppercase;
tt:n    text-transform:none;
tt:c    text-transform:capitalize;
tt:u    text-transform:uppercase;
tt:l    text-transform:lowercase;

```

```

tw       text-wrap:;
tw:n    text-wrap:normal;
tw:no   text-wrap:none;
tw:u    text-wrap:unrestricted;
tw:s    text-wrap:suppress;
tsh    text-shadow:hoff voff blur #000;
tsh:r   text-shadow:h v blur rgb(0, 0, 0);
tsh:ra  text-shadow:h v blur rgba(0, 0, 0,
      .5);
tsh+   text-shadow:0 0 0 #000;
tsh:n   text-shadow:none;
lh     line-height:;
lts    letter-spacing:;
lts:n  letter-spacing:normal;
whs    white-space:;
whs:n  white-space:normal;
whs:p   white-space:pre;
whs:nw  white-space:nowrap;
whs:pw  white-space:pre-wrap;
whs:pl  white-space:pre-line;
whsc   white-space-collapse:;
whsc:n  white-space-collapse:normal;
whsc:k  white-space-collapse:keep-all;

```

<b>Background</b>	
<b>bg</b>	background:#000;
<b>bg+</b>	background:#fff url() 0 0 no-repeat;
<b>bg:n</b>	background:none;
<b>bgc</b>	background-color:#fff;
<b>bgc:t</b>	background-color:transparent;
<b>bgi</b>	background-image:url();
<b>bgi:n</b>	background-image:none;
<b>bgr</b>	background-repeat:;
<b>bgr:n</b>	background-repeat:no-repeat;
<b>bgr:x</b>	background-repeat:repeat-x;
<b>bgr:y</b>	background-repeat:repeat-y;
<b>bgr:sp</b>	background-repeat:space;
<b>bgr:rd</b>	background-repeat:round;
<b>bga</b>	background-attachment:;
<b>bga:f</b>	background-attachment:fixed;
<b>bga:s</b>	background-attachment:scroll;
<b>bgp</b>	background-position:0 0;
<b>bgpx</b>	background-position-x:;
<b>bgpy</b>	background-position-y:;
<b>bgbk</b>	background-break:;
<b>bgbk:bb</b>	background-break:bounding-box;

```

bgbk:eb background-break:each-box;
bgbk:c background-break:continuous;
bgcp background-clip:padding-box;
bgcp:bb background-clip:border-box;
bgcp:pb background-clip:padding-box;
bgcp:cb background-clip:content-box;
bgcp:nc background-clip:no-clip;
bgo background-origin:;
bgo:pb background-origin:padding-box;
bgo:bb background-origin:border-box;
bgo:cb background-origin:content-box;
bgsz background-size:;
bgsz:a background-size:auto;
bgsz:ct background-size:contain;
bgsz:cv background-size:cover;

```

**Color**

```

c color:#000;
c:r color:rgb(0, 0, 0);
c:ra color:rgba(0, 0, 0, .5);
op opacity:;

Generated content

cnt content:'';
cnt:n, ct:n content:normal;
cnt:oq, ct:oq content:open-quote;
cnt:noq, ct:noq content:no-open-quote;
cnt:cq, ct:cq content:close-quote;
cnt:ncq, ct:ncq content:no-close-quote;
cnt:a, ct:a content:attr();
cnt:c, ct:c content:counter();
cnt:cs, ct:cs content:counters();
ct content:;

q quotes:;;
q:n quotes:none;
q:ru quotes:'\00AB' ,\00BB' ,\201E'
      ,\201C';
q:en quotes:'\201C' ,\201D' ,\2018'
      ,\2019';
coi counter-increment:;;
cor counter-reset:;;

```

## Outline

**ol** outline:;  
**ol:n** outline:none;  
**olo** outline-offset:;  
**olw** outline-width:;  
**olw:tn** outline-width:thin;  
**olw:m** outline-width:medium;  
**olw:tc** outline-width:thick;  
**ols** outline-style:;  
**ols:n** outline-style:none;  
**ols:dt** outline-style:dotted;  
**ols:ds** outline-style:dashed;  
**ols:s** outline-style:solid;  
**ols:db** outline-style:double;  
**ols:g** outline-style:groove;  
**ols:r** outline-style:ridge;  
**ols:i** outline-style:inset;  
**ols:o** outline-style:outset;  
**olc** outline-color:#000;  
**olc:i** outline-color:invert;

## Tables

**tbl** table-layout:;  
**tbl:a** table-layout:auto;  
**tbl:f** table-layout:fixed;  
**cps** caption-side:;  
**cps:t** caption-side:top;  
**cps:b** caption-side:bottom;  
**ec** empty-cells:;  
**ec:s** empty-cells:show;  
**ec:h** empty-cells:hide;

---

**Border**

**bd** border:;  
**bd+** border:1px solid #000;  
**bd:n** border:none;  
**bdk** border-break:close;  
**bdk:c** border-break:close;  
**bdcl** border-collapse:;  
**bdcl:c** border-collapse:collapse;  
**bdcl:s** border-collapse:separate;  
**bdc** border-color:#000;  
**bdc:t** border-color:transparent;  
**bdi** border-image:url();  
**bdi:n** border-image:none;

<b>bdti</b>	border-top-image:url();
<b>bdti:n</b>	border-top-image:none;
<b>bdri</b>	border-right-image:url();
<b>bdri:n</b>	border-right-image:none;
<b>bdbi</b>	border-bottom-image:url();
<b>bdbi:n</b>	border-bottom-image:none;
<b>bdli</b>	border-left-image:url();
<b>bdli:n</b>	border-left-image:none;
<b>bdc i</b>	border-corner-image:url();
<b>bdc i:n</b>	border-corner-image:none;
<b>bdc i:c</b>	border-corner-image:continue;
<b>bdtli</b>	border-top-left-image:url();
<b>bdtli:n</b>	border-top-left-image:none;
<b>bdtli:c</b>	border-top-left-image:continue;
<b>bdtri</b>	border-top-right-image:url();
<b>bdtri:n</b>	border-top-right-image:none;
<b>bdtri:c</b>	border-top-right-image:continue;
<b>bdbri</b>	border-bottom-right-image:url();
<b>bdbri:n</b>	border-bottom-right-image:none;
<b>bdbri:c</b>	border-bottom-right-image:continue;
<b>bdbli</b>	border-bottom-left-image:url();
<b>bdbli:n</b>	border-bottom-left-image:none;
<b>bdbli:c</b>	border-bottom-left-image:continue;
<b>bdf</b>	border-fit:repeat;
<b>bdf:c</b>	border-fit:clip;
<b>bdf:r</b>	border-fit:repeat;
<b>bdf:sc</b>	border-fit:scale;
<b>bdf:st</b>	border-fit:stretch;
<b>bdf:ow</b>	border-fit:overwrite;
<b>bdf:of</b>	border-fit:overflow;
<b>bdf:sp</b>	border-fit:space;
<b>bdlen</b>	border-length:;
<b>bdlen:a</b>	border-length:auto;
<b>bdsp</b>	border-spacing:;
<b>bds</b>	border-style:;
<b>bds:n</b>	border-style:none;
<b>bds:h</b>	border-style:hidden;
<b>bds:dt</b>	border-style:dotted;
<b>bds:ds</b>	border-style:dashed;
<b>bds:s</b>	border-style:solid;
<b>bds:db</b>	border-style:double;
<b>bds:dt�ds</b>	border-style:dot-dash;
<b>bds:dt�dts</b>	border-style:dot-dot-dash;
<b>bds:w</b>	border-style:wave;

**bds:g** border-style:groove;  
**bds:r** border-style:ridge;  
**bds:i** border-style:inset;  
**bds:o** border-style:outset;  
**bdw** border-width:;  
**bdt, bt** border-top:;  
**bdt+** border-top:1px solid #000;  
**bdt:n** border-top:none;  
**bdtw** border-top-width:;  
**bdts** border-top-style:;  
**bdts:n** border-top-style:none;  
**bdtc** border-top-color:#000;  
**bdtc:t** border-top-color:transparent;  
**bdr, br** border-right:;  
**bdr+** border-right:1px solid #000;  
**bdr:n** border-right:none;  
**bdrw** border-right-width:;  
**bdrst** border-right-style:;  
**bdrst:n** border-right-style:none;  
**bdrc** border-right-color:#000;  
**bdrc:t** border-right-color:transparent;  
**bdb, bb** border-bottom:;

**bdb+** border-bottom:1px solid #000;  
**bdb:n** border-bottom:none;  
**bdbw** border-bottom-width:;  
**bdb\$** border-bottom-style:;  
**bdbs:n** border-bottom-style:none;  
**bdbc** border-bottom-color:#000;  
**bdbc:t** border-bottom-color:transparent;  
**ndl, bl** border-left:;  
**ndl+** border-left:1px solid #000;  
**ndl:n** border-left:none;  
**ndlw** border-left-width:;  
**ndl\$** border-left-style:;  
**ndl\$:** border-left-style:none;  
**ndl\$:** border-left-color:#000;  
**ndl:t** border-left-color:transparent;  
**bdrs** border-radius:;  
**bdtrrs** border-top-right-radius:;  
**bdtlrs** border-top-left-radius:;  
**bdbrrs** border-bottom-right-radius:;  
**bdbllrs** border-bottom-left-radius:;

**Lists**

**lis** list-style:;;  
**lis:n** list-style:none;  
**lisp** list-style-position:;;  
**lisp:i** list-style-position:inside;  
**lisp:o** list-style-position:outside;  
**list** list-style-type:;;  
**list:n** list-style-type:none;  
**list:d** list-style-type:disc;  
**list:c** list-style-type:circle;  
**list:s** list-style-type:square;  
**list:dc** list-style-type:decimal;  
**list:dclz** list-style-type:decimal-leading-zero;  
**list:lr** list-style-type:lower-roman;  
**list:ur** list-style-type:upper-roman;  
**lisi** list-style-image:;;  
**lisi:n** list-style-image:none;

**Print**

**pgbb** page-break-before:;;  
**pgbb:au** page-break-before:auto;  
**pgbb:al** page-break-before:always;  
**pgbb:l** page-break-before:left;

**pgbb:r** page-break-before:right;  
**pgbi** page-break-inside:;;  
**pgbi:au** page-break-inside:auto;  
**pgbi:av** page-break-inside:avoid;  
**pgba** page-break-after:;;  
**pgba:au** page-break-after:auto;  
**pgba:al** page-break-after:always;  
**pgba:l** page-break-after:left;  
**pgba:r** page-break-after:right;  
**orp** orphans:;;  
**wid** widows:;;

**Others**

**!** !important  
**@f** @font-face {  
 font-family:;;  
 src:url(|);  
}  
**@f+** @font-face {  
 font-family: ,FontName';  
 src: url(,FileName.eot');  
 src: url(,FileName.eot#iefix')  
 format(embedded-opentype'),  
 url(,FileName.woff')  
 format(woff'),  
 url(,FileName.ttf')  
 format(truetype'),  
 url(,FileName.  
 svg#FontName') format(svg');  
 font-style: normal;  
 font-weight: normal;  
}

```

@i, @import
  @import url();

@kf
  @-webkit-keyframes identifier {
    from { }
    to { }
  }
  @-o-keyframes identifier {
    from { }
    to { }
  }
  @-moz-keyframes identifier {
    from { }
    to { }
  }
  @keyframes identifier {
    from { }
    to { }
  }

@m, @media
  @media screen { }

ac
  align-content:;

ac:c
  align-content:center;

ac:fe
  align-content:flex-end;

ac:fs
  align-content:flex-start;

ac:s
  align-content:stretch;

ac:sa
  align-content:space-around;

ac:sb
  align-content:space-between;

ai
  align-items:;

ai:b
  align-items:baseline;

ai:c
  align-items:center;

ai:fe
  align-items:flex-end;

ai:fs
  align-items:flex-start;

ai:s
  align-items:stretch;

```

Animation	
<b>anim</b>	animation:;
<b>anim-</b>	animation:name duration timing-function delay iteration-count direction fill-mode;
<b>animdel</b>	animation-delay:time;
<b>animdir</b>	animation-direction:normal;
<b>animdir:a</b>	animation-direction:alternate;
<b>animdir:ar</b>	animation-direction:alternate-reverse;
<b>animdir:n</b>	animation-direction:normal;
<b>animdir:r</b>	animation-direction:reverse;
<b>animdur</b>	animation-duration:0s;
<b>animfm</b>	animation-fill-mode:both;
<b>animfm:b</b>	animation-fill-mode:backwards;
<b>animfm:bt, animfm:bh</b>	animation-fill-mode:both;
<b>animfm:f</b>	animation-fill-mode:forwards;
<b>animic</b>	animation-iteration-count:1;
<b>animic:i</b>	animation-iteration-count:infinite;
<b>animn</b>	animation-name:none;
<b>animps</b>	animation-play-state:running;
<b>animps:p</b>	animation-play-state:paused;
<b>animps:r</b>	animation-play-state:running;
<b>animtf</b>	animation-timing-function:linear;

```

animtf:cb
  animation-timing-function:cubic-bezier(0.1, 0.7, 1.0, 0.1);

animtf:e
  animation-timing-function:ease;

animtf:ei
  animation-timing-function:ease-in;

animtf:eo
  animation-timing-function:ease-in-out;

animtf:eo
  animation-timing-function:ease-out;

animtf:l
  animation-timing-function:linear;

ap
  appearance:${none};

as
  align-self:;

as:a
  align-self:auto;

as:b
  align-self:baseline;

as:c
  align-self:center;

as:fe
  align-self:flex-end;

as:fs
  align-self:flex-start;

as:s
  align-self:stretch;

bfv
  backface-visibility:;

bfv:h
  backface-visibility:hidden;

bfv:v
  backface-visibility:visible;

bg:ie
  filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(src='x.png',sizingMethod='crop');

cm
  /* ${child} */

colm
  columns:;

colmc
  column-count:;

```

```

colmf
  column-fill:;

colmg
  column-gap:;

colmr
  column-rule:;

colmrc
  column-rule-color:;

colmrs
  column-rule-style:;

colmrw
  column-rule-width:;

colms
  column-span:;

colmw
  column-width:;

d:ib+
  display: inline-block;
  *display: inline;
  *zoom: 1;

```

## Flex

```

fx
  flex:;

fxb
  flex-basis:;

fxd
  flex-direction:;

fxd:c
  flex-direction:column;

fxd:cr
  flex-direction:column-reverse;

fxd:r
  flex-direction:row;

fxd:rr
  flex-direction:row-reverse;

fxf
  flex-flow:;

fxg
  flex-grow:;

fxsh
  flex-shrink:;

fxw
  flex-wrap: ;

```

	Transform, Transition
<b>fxw:n</b>	flex-wrap:nowrap;
<b>fxw:w</b>	flex-wrap:wrap;
<b>fxw:wr</b>	flex-wrap:wrap-reverse;
<b>jc</b>	justify-content:;
<b>jc:c</b>	justify-content:center;
<b>jc:fe</b>	justify-content:flex-end;
<b>jc:fs</b>	justify-content:flex-start;
<b>jc:sa</b>	justify-content:space-around;
<b>jc:sb</b>	justify-content:space-between;
<b>mar</b>	max-resolution:res;
<b>mir</b>	min-resolution:res;
<b>op+</b>	opacity: ; filter: alpha(opacity=);
<b>op:ie</b>	filter:progid:DXImageTransform.Microsoft.Alpha(Opacity=100);
<b>op:ms</b>	-ms-filter:'progid:DXImageTransform.Microsoft.Alpha(Opacity=100)';
<b>ord</b>	order:;
<b>ori</b>	orientation:;
<b>ori:l</b>	orientation:landscape;
<b>ori:p</b>	orientation:portrait;
<b>tov</b>	text-overflow:{ellipsis};
<b>tov:c</b>	text-overflow:clip;
<b>tov:e</b>	text-overflow:ellipsis;
<b>trf</b>	transform:;
<b>trf:r</b>	transform: rotate(angle);
<b>trf:rx</b>	transform: rotateX(angle);
<b>trf:ry</b>	transform: rotateY(angle);
<b>trf:rz</b>	transform: rotateZ(angle);
<b>trf:sc</b>	transform: scale(x, y);
<b>trf:sc3</b>	transform: scale3d(x, y, z);
<b>trf:scx</b>	transform: scaleX(x);
<b>trf:scy</b>	transform: scaleY(y);
<b>trf:scz</b>	transform: scaleZ(z);
<b>trf:skx</b>	transform: skewX(angle);
<b>trf:sky</b>	transform: skewY(angle);
<b>trf:t</b>	transform: translate(x, y);
<b>trf:t3</b>	transform: translate3d(tx, ty, tz);
<b>trf:tx</b>	transform: translateX(x);
<b>trf:ty</b>	transform: translateY(y);
<b>trf:tz</b>	transform: translateZ(z);
<b>trfo</b>	transform-origin:;
<b>trfs</b>	transform-style:preserve-3d;
<b>trs</b>	transition:prop time;
<b>trsde</b>	transition-delay:time;

**XSL**

```

trsdu          transition-duration:time;
trsp          transition-property:prop;
trstf         transition-timing-function:tfunc;
us            user-select:${none};
wfsm          -webkit-font-
                  smoothing:${antialiased};
wfsm:a        -webkit-font-smoothing:antialiased;
wfsm:n        -webkit-font-smoothing:none;
wfsm:s, wfsm:sa -webkit-font-smoothing:subpixel-
                  antialiased;
wm            writing-mode:lr-tb;
wm:btl        writing-mode:bt-lr;
wm:btr        writing-mode:bt-rl;
wm:lrb        writing-mode:lr-bt;
wm:lrt        writing-mode:lr-tb;
wm:rlb        writing-mode:rl-bt;
wm:rlt        writing-mode:rl-tb;
wm:tbl        writing-mode:tb-lr;
wm:tbr        writing-mode:tb-rl;

```

```

tmatch, tm    <xsl:template match="" mode=""></
                  xsl:template>
tname, tn    <xsl:template name=""></
                  xsl:template>
call          <xsl:call-template name="" />
ap            <xsl:apply-templates select="" 
                  mode="" />
api          <xsl:apply-imports />
imp          <xsl:import href="" />
inc          <xsl:include href="" />
ch            <xsl:choose></xsl:choose>
xsl:when, wh <xsl:when test=""></xsl:when>
ot            <xsl:otherwise></xsl:otherwise>
if            <xsl:if test=""></xsl:if>
par          <xsl:param name=""></xsl:param>
pare         <xsl:param name="" select="" />
var          <xsl:variable name=""></
                  xsl:variable>
vare         <xsl:variable name="" select="" />
wp            <xsl:with-param name="" select="" />
key          <xsl:key name="" match="" use="" />
elem         <xsl:element name=""></xsl:element>
attr         <xsl:attribute name=""></
                  xsl:attribute>
attrs        <xsl:attribute-set name=""></
                  xsl:attribute-set>
cp           <xsl:copy select="" />

```

## HTML DOCTYPES

```

co
  <xsl:copy-of select="" />
val
  <xsl:value-of select="" />
each, for
  <xsl:for-each select=""></
  xsl:for-each>
tex
  <xsl:text></xsl:text>
com
  <xsl:comment></xsl:comment>
msg
  <xsl:message terminate="no"></
  xsl:message>
fall
  <xsl:fallback></xsl:fallback>
num
  <xsl:number value="" />
nam
  <namespace-alias stylesheet-prefix="">
  result-prefix="" />
pres
  <xsl:preserve-space elements="" />
strip
  <xsl:strip-space elements="" />
proc
  <xsl:processing-
  instruction name=""></
  xsl:processing-instruction>
sort
  <xsl:sort select="" order="" />
choose+
  <xsl:choose>
    <xsl:when test=""></xsl:when>
    <xsl:otherwise></xsl:otherwise>
  </xsl:choose>
xsl
  {}
  <?xml version="1.0"
  encoding="UTF-8"?>
  <xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/
  XSL/Transform"></xsl:stylesheet>
!!!
  <?xml version="1.0"
  encoding="UTF-8"?>

```

### !, html:5

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>Document</title>
</head>
<body>

</body>
</html>

```

### doc

```

<html>
<head>
  <meta charset="UTF-8" />
  <title>Document</title>
</head>
<body>

</body>
</html>

```

### doc4

```

<html>
<head>
  <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8"
/>
  <title>Document</title>
</head>
<body>

</body>
</html>

```

### html:4t

```

<!DOCTYPE HTML PUBLIC "-//W3C//"
DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.
dtd">
<html lang="en">
<head>
  <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8"
/>
  <title>Document</title>
</head>
<body>

</body>
</html>

```

**html:4s**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD
HTML 4.01//EN" "http://www.w3.org/
TR/html4/strict.dtd">
<html lang="en">
<head>
  <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8"
/>
  <title>Document</title>
</head>
<body>

</body>
</html>
```

**html:xt**

```
<!DOCTYPE html PUBLIC "-//W3C//"
DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd">
<html xmlns="http://www.
w3.org/1999/xhtml" xml:lang="en">
<head>
  <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8"
/>
  <title>Document</title>
</head>
<body>

</body>
</html>
```

**html:xs**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD
XHTML 1.0 Strict//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-strict.
dtd">
<html xmlns="http://www.
w3.org/1999/xhtml" xml:lang="en">
<head>
  <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8"
/>
  <title>Document</title>
</head>
<body>

</body>
</html>
```

**html:xxs**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD
XHTML 1.1//EN" "http://www.w3.org/
TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.
w3.org/1999/xhtml" xml:lang="en">
<head>
  <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8"
/>
  <title>Document</title>
</head>
<body>

</body>
</html>
```

!!!

```
<!DOCTYPE html>
```

**!!!4t**

```
<!DOCTYPE HTML PUBLIC "-//W3C//"
DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.
dtd">
```

**!!!4s**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD
HTML 4.01//EN" "http://www.w3.org/
TR/html4/strict.dtd">
```

**!!!xt**

```
<!DOCTYPE html PUBLIC "-//W3C//"
DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd">
```

**!!!xs**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD
XHTML 1.0 Strict//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-strict.
dtd">
```

**!!!xxs**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD
XHTML 1.1//EN" "http://www.w3.org/
TR/xhtml11/DTD/xhtml11.dtd">
```