|  |  |
| --- | --- |
| Syntax | |
| /\* Comments \*/ *Note:*  @media type { *media type is optional*  selector {  property: value;  }  } | |
| Inline Style  <tag style="property: value;"> | |
| Embedded Style  <head>  <style type="text/css">  selector { property: value; }  </style>  </head> | |
| External Style Sheet  <head>  <link rel="stylesheet"  padding-top padding-right  padding-bottom padding-left  padding  margin-top margin-right  margin-bottom margin-left  margin  Boxes  type="text/css" href="style.css" />  </head> | |
| Selectors | |
| \* | All elements |
| tag | All tag elements |
| tag \* | All elements within tag |
| tag tag2 | border-top-width border-right-width  border-bottom-width border-left-width  border-width  border-top-style border-right-style  border-bottom-style border-left-style  border-style  border-top-color border-right-color  border-bottom-color border-left-color  border-color  border-top border-right  border-bottom border-left  border  tag2 elements (any childhood level ) within tag |
| tag, tag2 | All tag and tag2 elements |
| tag > tag2 | tag2 is a direct child of tag |
| tag + tag2 | tag2 preceded by tag |
| .class | Elements with class ‘class’ |
| tag.class | All tags with class ‘class’ |
| #id | Element with id ‘id’ |
| tag#id | Tag with id ‘id’ |

Box Model

|  |  |  |
| --- | --- | --- |
| Size and colors | | |
| Relative sizes | em | rem | ex | ch  vw | vh | vmin | vmax | % | |
| Absolute sizes | cm | mm | in | px | pt | pc | |
| Colors | Hex #ff00ff  RGB rgb(255,0,255) | |
| Positioning | | |
| position | absolute | fixed | relative | static | inherit | |
| float | left | right | none | inherit | |
| top, right,  bottom, left | Sets the offset from the edge | |
| clear | left | right | both | none | inherit I fixed the document now | |
| display | Specifies element placing in the document flow | |
| overflow | visible | hidden | scroll | auto | inherit | |
| visibility | visible | hidden | collapse | inherit | |
| z-index | auto | int | inherit | |
| Dimensions | | |
| height | | auto | int | % | inherit |
| max-height | | none | int | % | inherit |
| max-height | | none | int | % | inherit |
| max-width | | none | int | % | inherit |
| min-height | | int | % | inherit |
| min-width | | int | % | inherit |
| width | | auto | int | % | inherit |
| Other | | |
| background | | Format:  background (color) (image) (repeat) (position) |
| cursor | | Set the type of cursor |
| quotes | | Set type of quotation marks |

|  |  |
| --- | --- |
| Pseudo-selectors | |
| :active | Adds style to active element |
| :after | Adds content after element |
| :before | Ads content before element |
| : first-child | Adds style to first child |
| : first-letter | Adds style to first character |
| : first-line | Adds style to first line |
| :focus | Adds style to focused element |
| :hover | Adds style when mouse is over |
| :link | Adds style to unvisited link |
| :visited | Adds style to visited link |
| Text | |
| font-family | Specifies the font family |
| font-size | Specifies the font size |
| font-style | normal | italic | oblique | inherit |
| font-variant | normal | small-caps | inherit |
| font-weight | normal | bold | bolder | lighter | int (100- 900) | inherit |
| color | Sets the color of text |
| line-height | normal | int | % | inherit |
| text-align | left | right | center | justify | inherit |
| text-  decoration | none | underline | overline | line-through | blink | inherit |
| text-indent | int | % | inherit |
| text-  transform | none | capitalize | uppercase | lowercase | inherit |
| vertical  -align | int | % | baseline | sub | super | top | text-top  middle | bottom | text-bot- tom | inherit |
| white  -space | normal | nowrap | pre |  pre-line | pre-wrap| inherit |
| word-  spacing | normal | length | inherit |

**CSS Precedence Rules**

When the browser needs to resolve what styles to apply to a given HTML element, it uses a set of CSS precedence rules. Given these rules, the browser can determine what styles to apply. The rules are:

1. !important after CSS properties.
2. Specificity of CSS rule selectors.
3. Sequence of declaration.

Note, that CSS precedence happens at CSS property level. Thus, if two CSS rules target the same HTML element, and the first CSS rule takes precedence over the second, then all CSS properties specified in the first CSS rule takes precedence over the CSS properties declared in the second rule. However, if the second CSS rule contains CSS properties that are not specified in the first CSS rule, then these are still applied. The CSS rules are combined - not overriding each other.

The different CSS selector types has different specificity. By specificity is meant how specifically the CSS selector targets the element is selects. Here is a list of CSS selector specificity:

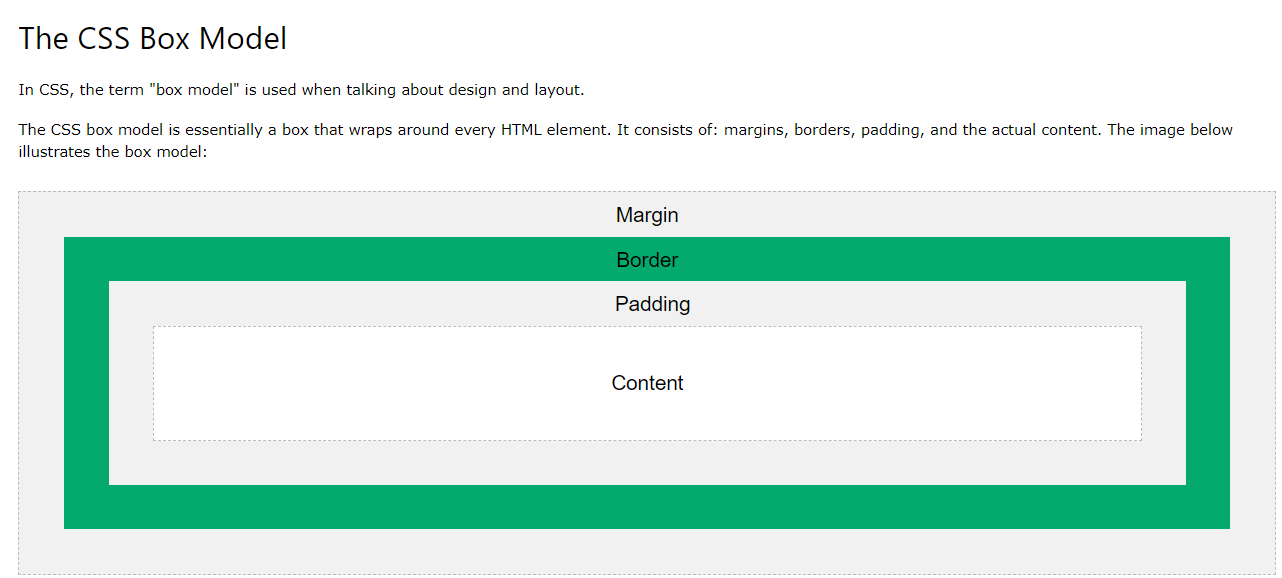
|  |  |
| --- | --- |
| **CSS Selector** | **Description** |
| Inherited styles | Lowest specificity of all selectors - since an inherited style targets the element's parent, and not the HTML element itself. |
| \* | Lowest specificity of all directly targeted selectors |
| element | Higher specificity than universal selector and inherited styles. |
| attribute | Higher specificity than element selector |
| Class | Higher specificity than attribute, element and universal selectors. |
| ID | Higher specificity than class selector. |
| Combined selectors | Gets the specificity of the selectors combined. |
| CSS properties set directly on element, inside style attribute. | Stronger specificity than ID selector. |

CSS selectors:

|  |  |  |
| --- | --- | --- |
| **Selector** | **Example** | **selects** |
| [.*class*](https://www.w3schools.com/cssref/sel_class.asp) | .intro | all tags with class="intro" |
| .class1.class2 | .name1.name2 | all tags with both name1 and name2 set within its class attribute |
| .class1 .class2 | .name1 .name2 | all tags with name2 that is a descendant of an tag with name1 |
| [#*id*](https://www.w3schools.com/cssref/sel_id.asp) | #firstname | the tag with id="firstname" |
| [\*](https://www.w3schools.com/cssref/sel_all.asp) | \* | all tags |
| [*tag*](https://www.w3schools.com/cssref/sel_element.asp) | p | all <p> tags |
| [*tag.class*](https://www.w3schools.com/cssref/sel_element_class.asp) | p.intro | all <p> tags with class="intro" |
| [*tag,tag*](https://www.w3schools.com/cssref/sel_element_comma.asp) | div, p | all <div> tags and all <p> tags |
| [*tag* *tag*](https://www.w3schools.com/cssref/sel_element_element.asp) | div p | all <p> tags inside <div> tags |
| [*tag*>*tag*](https://www.w3schools.com/cssref/sel_element_gt.asp) | div > p | all <p> tags where the parent is a <div> tag |
| [*tag*+*tag*](https://www.w3schools.com/cssref/sel_element_pluss.asp) | div + p | the first <p> tag that is placed immediately after <div> tags |
| [*tag1*~*tag2*](https://www.w3schools.com/cssref/sel_gen_sibling.asp) | p ~ ul | every <ul> tag that is preceded by a <p> tag |
| [[*attribute*]](https://www.w3schools.com/cssref/sel_attribute.asp) | [target] | all tags with a target attribute |
| [[*attribute*=*value*]](https://www.w3schools.com/cssref/sel_attribute_value.asp) | [target=\_blank] | all tags with target="\_blank" |
| [[*attribute*~=*value*]](https://www.w3schools.com/cssref/sel_attribute_value_contains.asp) | [title~=flower] | all tags with a title attribute containing the word "flower" |
| [[*attribute*|=*value*]](https://www.w3schools.com/cssref/sel_attribute_value_lang.asp) | [lang|=en] | all tags with a lang attribute value starting with "en" |
| [[*attribute*^=*value*]](https://www.w3schools.com/cssref/sel_attr_begin.asp) | a[href^="https"] | every <a> tag whose href attribute value begins with "https" |
| [[*attribute*$=*value*]](https://www.w3schools.com/cssref/sel_attr_end.asp) | a[href$=".pdf"] | every <a> tag whose href attribute value ends with ".pdf" |
| [[*attribute*\*=*value*]](https://www.w3schools.com/cssref/sel_attr_contain.asp) | a[href\*="w3schools"] | every <a> tag whose href attribute value contains the substring "w3schools" |
| [:active](https://www.w3schools.com/cssref/sel_active.asp) | a:active | the active link |
| [::after](https://www.w3schools.com/cssref/sel_after.asp) | p::after | Insert something after the content of each <p> tag |
| [::before](https://www.w3schools.com/cssref/sel_before.asp) | p::before | Insert something before the content of each <p> tag |
| [:checked](https://www.w3schools.com/cssref/sel_checked.asp) | input:checked | every checked <input> tag |
| [:default](https://www.w3schools.com/cssref/sel_default.asp) | input:default | the default <input> tag |
| [:disabled](https://www.w3schools.com/cssref/sel_disabled.asp) | input:disabled | every disabled <input> tag |
| [:empty](https://www.w3schools.com/cssref/sel_empty.asp) | p:empty | every <p> tag that has no children (including text nodes) |
| [:enabled](https://www.w3schools.com/cssref/sel_enabled.asp) | input:enabled | every enabled <input> tag |
| [:first-child](https://www.w3schools.com/cssref/sel_firstchild.asp) | p:first-child | every <p> tag that is the first child of its parent |
| [::first-letter](https://www.w3schools.com/cssref/sel_firstletter.asp) | p::first-letter | the first letter of every <p> tag |
| [::first-line](https://www.w3schools.com/cssref/sel_firstline.asp) | p::first-line | the first line of every <p> tag |
| [:first-of-type](https://www.w3schools.com/cssref/sel_first-of-type.asp) | p:first-of-type | every <p> tag that is the first <p> tag of its parent |
| [:focus](https://www.w3schools.com/cssref/sel_focus.asp) | input:focus | the input tag which has focus |
| [:fullscreen](https://www.w3schools.com/cssref/sel_fullscreen.asp) | :fullscreen | the tag that is in full-screen mode |
| [:hover](https://www.w3schools.com/cssref/sel_hover.asp) | a:hover | links on mouse over |
| [:in-range](https://www.w3schools.com/cssref/sel_in-range.asp) | input:in-range | input tags with a value within a specified range |
| [:indeterminate](https://www.w3schools.com/cssref/sel_indeterminate.asp) | input:indeterminate | input tags that are in an indeterminate state |
| [:invalid](https://www.w3schools.com/cssref/sel_invalid.asp) | input:invalid | all input tags with an invalid value |
| [:lang(*language*)](https://www.w3schools.com/cssref/sel_lang.asp) | p:lang(it) | every <p> tag with a lang attribute equal to "it" (Italian) |
| [:last-child](https://www.w3schools.com/cssref/sel_last-child.asp) | p:last-child | every <p> tag that is the last child of its parent |
| [:last-of-type](https://www.w3schools.com/cssref/sel_last-of-type.asp) | p:last-of-type | every <p> tag that is the last <p> tag of its parent |
| [:link](https://www.w3schools.com/cssref/sel_link.asp) | a:link | all unvisited links |
| [::marker](https://www.w3schools.com/cssref/sel_marker.asp) | ::marker | the markers of list items |
| [:not(*selector*)](https://www.w3schools.com/cssref/sel_not.asp) | :not(p) | every tag that is not a <p> tag |
| [:nth-child(*n*)](https://www.w3schools.com/cssref/sel_nth-child.asp) | p:nth-child(2) | every <p> tag that is the second child of its parent |
| [:nth-last-child(*n*)](https://www.w3schools.com/cssref/sel_nth-last-child.asp) | p:nth-last-child(2) | every <p> tag that is the second child of its parent, counting from the last child |
| [:nth-last-of-type(*n*)](https://www.w3schools.com/cssref/sel_nth-last-of-type.asp) | p:nth-last-of-type(2) | every <p> tag that is the second <p> tag of its parent, counting from the last child |
| [:nth-of-type(*n*)](https://www.w3schools.com/cssref/sel_nth-of-type.asp) | p:nth-of-type(2) | every <p> tag that is the second <p> tag of its parent |
| [:only-of-type](https://www.w3schools.com/cssref/sel_only-of-type.asp) | p:only-of-type | every <p> tag that is the only <p> tag of its parent |
| [:only-child](https://www.w3schools.com/cssref/sel_only-child.asp) | p:only-child | every <p> tag that is the only child of its parent |
| [:optional](https://www.w3schools.com/cssref/sel_optional.asp) | input:optional | input tags with no "required" attribute |
| [:out-of-range](https://www.w3schools.com/cssref/sel_out-of-range.asp) | input:out-of-range | input tags with a value outside a specified range |
| [::placeholder](https://www.w3schools.com/cssref/sel_placeholder.asp) | input::placeholder | input tags with the "placeholder" attribute specified |
| [:read-only](https://www.w3schools.com/cssref/sel_read-only.asp) | input:read-only | input tags with the "readonly" attribute specified |
| [:read-write](https://www.w3schools.com/cssref/sel_read-write.asp) | input:read-write | input tags with the "readonly" attribute NOT specified |
| [:required](https://www.w3schools.com/cssref/sel_required.asp) | input:required | input tags with the "required" attribute specified |
| [:root](https://www.w3schools.com/cssref/sel_root.asp) | :root | the document's root tag |
| [::selection](https://www.w3schools.com/cssref/sel_selection.asp) | ::selection | the portion of an tag that is selected by a user |
| [:target](https://www.w3schools.com/cssref/sel_target.asp) | #news:target | the current active #news tag (clicked on a URL containing that anchor name) |
| [:valid](https://www.w3schools.com/cssref/sel_valid.asp) | input:valid | all input tags with a valid value |

Example on ::after selector





**Font styling:**

When the text-align property is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight, In CSS there are five generic font families:

1. **Serif** fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.
2. **Sans-serif** fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.
3. **Monospace** fonts - here all the letters have the same fixed width. They create a mechanical look.
4. **Cursive** fonts imitate human handwriting.
5. **Fantasy** fonts are decorative/playful fonts.



Font Pairing Rules

1. Compliment 🡪 It is always safe to find font pairings that complement one another.

2. Use Font Superfamilies 🡪 A font superfamily is a set of fonts designed to work well together. So, using different fonts within the same superfamily is safe.

3. Contrast is King 🡪 Two fonts that are too similar will often conflict. However, contrasts, done the right way, brings out the best in each font.

4. Choose Only One Boss

**The position Property**

The position property specifies the type of positioning method used for an element.

* Static
  + An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:
* Relative
  + Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.
* Fixed
  + An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.
* Absolute
  + An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).
  + However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.
* Sticky
  + An element with position: sticky; is positioned based on the user's scroll position.
  + A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).

**The overflow Property**

The CSS overflow property controls what happens to content that is too big to fit into an area, it specifies whether to clip the content or to add scrollbars when the content of an element is too big to fit in the specified area.

* visible - Default. The overflow is not clipped. The content renders outside the element's box
* hidden - The overflow is clipped, and the rest of the content will be invisible
* scroll - The overflow is clipped, and a scrollbar is added to see the rest of the content
* auto - Similar to scroll, but it adds scrollbars only when necessary

**The clear Property**

When we use the float property, and we want the next element below (not on right or left), we will have to use the clear property, The clear property specifies what should happen with the element that is next to a floating element.

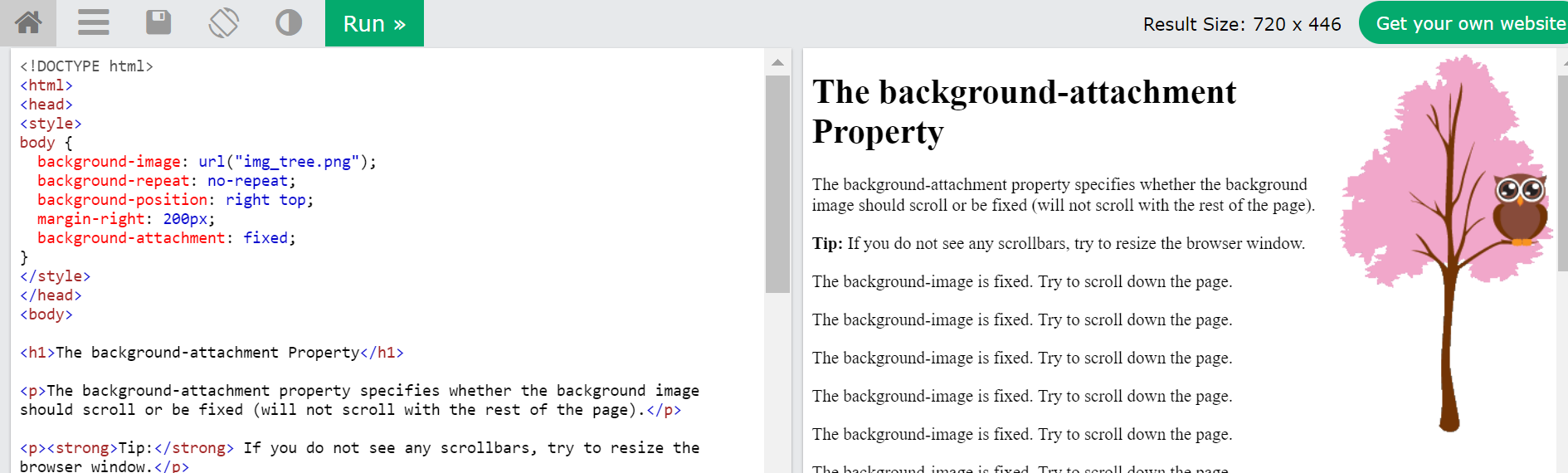
* none - The element is not pushed below left or right floated elements. This is default
* left - The element is pushed below left floated elements
* right - The element is pushed below right floated elements
* both - The element is pushed below both left and right floated elements
* inherit - The element inherits the clear value from its parent

## **Block-level Elements:** A block-level element always starts on a new line and takes up the full width available, <div> , <h1> - <h6> , <p> , <form>

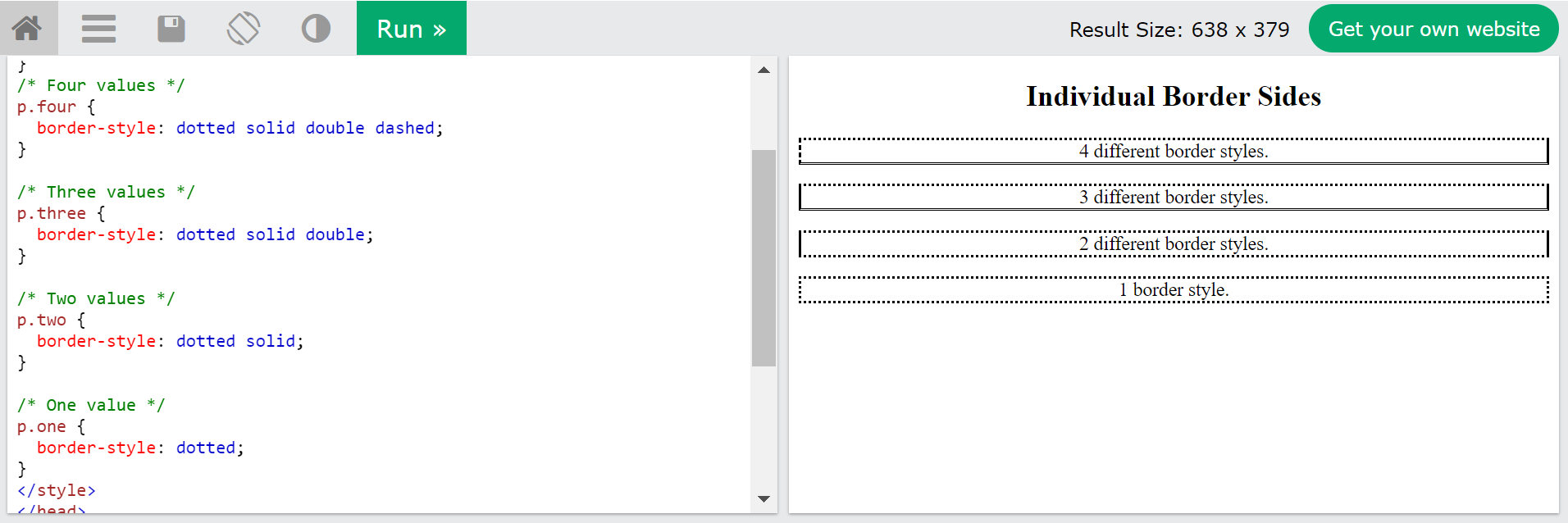
## **Inline Elements:** An inline element does not start on a new line and only takes up as much width as necessary. <span> , <a> , <img>

**Styling examples**

Defining fixed background:



Border sides:



list-style-position:

