Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for contents of text-input area

Checker Input

Show [x] source [ ] outline [ ] image report Options…

Check by text input [x] CSS

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <meta name="author" content="pf">
    <link rel="stylesheet" type="text/css" href="CSS_Template/style.css">
</head>
<body>
    <div class="content-width">
    <header>
        <h1 id="top">Flexbox</h1>
    </header>
    </div>
```

Check

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

Document checking completed. No errors or warnings to show.

Source

```
1.  <!DOCTYPE html>↩
2.  <html lang="en">↩
3.  <head>↩
4.      <meta charset="UTF-8">↩
5.      <meta name="viewport" content="width=device-width, initial-scale=1.0">↩
6.      <title>Document</title>↩
7.      <meta name="author" content="pf">↩
8.      <link rel="stylesheet" type="text/css" href="CSS_Template/style.css">↩
9.  </head>↩
10. <body>↩
11.     <div class="content-width">↩
12.     <header>↩
13.         <h1 id="top">Flexbox</h1>↩
14.     </header>↩
15.     </div>↩
```

```
16.     <nav class="menu" id="main-menu">↩
17.         <button class="menu-toggle" id="toggle-menu">↩
18.             toggle menu↩
19.         </button>↩
20.         <div class="menu-dropdown">↩
21.             <ul class="nav-menu">↩
22.                 <li><a href="#section1">What is Flexbox</a></li>↩
23.                 <li><a href="#section2">History of Flexbox</a></li>↩
24.                 <li><a href="#section3">Flexbox Parent Properties</a></li>↩
25.                 <li><a href="#section4">Flexbox Child Properties</a></li>↩
26.                 <!-- <li><a href="#section5">Hello</a></li> -->↩
27.             </ul>↩
28.         </div>↩
29.     </nav>↩
30.     <div class="content-width">↩
31.     <main>↩
32.         <section class="section1" id="section1">↩
33.             <h2>What is Flexbox?</h2>↩
34.             <p class="paragraph">Flexbox is an abbreviation of Flexible Box
    Layout Module. It is used to arrange items on the web page into rows or columns.
    Flexbox allows for flexible website layout designs with greater precision than
    what can be done with methods like float or positioning. Flexbox is designed
    more for when you need a one-dimensional layout, unlike grid layout which is
    ment for two-dimensional layouts. Before Flexbox was created, there were four
    different modes to use in website layouts. Those modes that preceded it were
    block, inline, table, and positioned. </p>↩
35.             <p class="paragraph">The reason that flexbox was created was to make
    a layout that was more efficient in how it aligned and distributed space between
    the items in its container, especially in cases where the exact size of the
    items in the container are either unknown or dynamic. Flexbox enables the items
    inside it to be expanded to fit available free space or shrinks the items to
    prevent them from overflowing. This is something that was revolutionary when it
    was released as previous layouts that were used for websites were unable to
    expand or shrink items like Flexbox can. The Flexbox layout was also
    revolutionary as all previous layouts for websites had their own inherent
    direction they were required to go in. This is different from how flexbox works
    as it's able to organize the items in either a vertical or horizontal
    orientation, allowing for more design options. Another thing that differentiated
    flexbox from some of the previous layout models was that it is purely a part of
    css, which is unlike the older model that flexbox is most like, the block model,
    which is in both html and css. These many differences are what made flexbox as
    important to the design of websites that it has become, though the process of
    its creation is also important to understanding this website layout model.↩
36.             </p>↩
37.             <img src="Assets/Flexbox.JPG" alt="Flexbox">↩
38.         </section>↩
39.         <section class="section2" id="section2">↩
40.             <h2>History of Flexbox</h2>↩
41.                 <p class="paragraph">The original idea that flexbox came from
    was first proposed in 2008 by the CSS Working Group, with the first working
    draft of it being finished on July 23rd, 2009. The original design was based on
    technology used by the browser Firefox called XUL, which Firefox used to make
    their user interface. However the algorithm of the original design was slow and
    worked differently depending on the implementation. This led to flexbox having
    its original specifications to be overhauled by Tab Atkins Jr in 2011, who made
    two different working versions as drafts. The unofficial syntax that he used in
    these versions got supported by some browsers like Chrome and Internet Explorer
    10. The 2012 version of it that he had helped to create was put up for W3C
    candidate recommendation, which means that the specification was stable, that it
    was supported by browsers, and that tests are being written for it, with major
    changes to it being unlikely at this point.The main goal Tab Atkins had in
    rebuilding how flex works was to get rid of the dependencies that were on
    everything and were very common at the time. ↩
42.                 </p>↩
43.                     <img src="Assets/firefox.png" alt="Firefox Logo">↩
```

```
                <p class="paragraph">Tab Atkins went on to rewrite the editor’s draft of it in 2013 to allow it to work better with some new specifications. The changes that he made to flex allowed for it to have an improved algorithm as well as giving the system greater efficiency than it had before. The last draft of flexbox was published in May of 2015 and was put up for review the month after. Finally, the W3C officially added it to the W3C Candidate Recommendations during October of 2017. Flexbox is now an increasingly common part of website development due to how helpful it can be in making a modern design for a website.</p>↩
                <img src="Assets/Flexbox_timeline.JPG" alt="Flexbox Timeline">↩
        </section>↩
        <section  class="section3" id="section3">↩
            <h2>Flexbox Parent Properties</h2>↩
            <div class="container">↩
                    <p>Parent properties are properties that affect everything inside the flexbox.↩
                    </p>↩
                <h3>Display</h3>↩
                    <p>This defines that something is in a flexbox and enables flex content for any children it has. Flexbox can’t be used without setting up the display first.</p>↩
                    <img src="Assets/Display_Flex.JPG" alt="Display Flex">↩
                <h3>Flex-Direction</h3>↩
                    <p>This establishes the direction that the flex will be going through defining the main axis of orientation. It also allows you to have the flexbox go in reverse along the main axis of orientation. The direction is determined by row, row-reverse, column, and column-reverse.↩
                    </p>↩
                    <img src="Assets/Flex_Direction.JPG" alt="Flex-direction">↩
                <h3>Flex-Wrap</h3>↩
                    <p>This allows items to wrap around as well as the direction of the wrap. The default setting of Flex-wrap is nowrap, meaning it will try to fit all the content onto one line. The options for this setting are nowrap, wrap, and wrap-reverse.↩
                    </p>↩
                    <img src="Assets/Flex_Wrap.JPG" alt="Flex-wrap">↩
                <h3>Flex-Flow</h3>↩
                    <p>This is a shorthand that combines Flex-direction with Flex-wrap, allowing someone to code both on the same line of code. Flex-Flow uses the key words of row, row-reverse, column, column-reverse, nowrap, wrap, and wrap-reverse.↩
                    </p>↩
                    <img src="Assets/Flex_Flow.JPG" alt="Flex-flow">↩
                <h3>Justify-Content</h3>↩
                    <p>This defines how the items will be aligned on the main axis, especially how much space is between items. It also has some amount of control over how items align when they overflow the line. The key words for it are flex-start, flex-end, start, end, left, right, center, space-between, space-around, and space-evenly.↩
                    </p>↩
                    <img src="Assets/Justify_Content.JPG" alt="Justify-content">↩
                <h3>Align-Items</h3>↩
                    <p>This works like the justify-content of the cross-axis, with it defining what the orientation of items is laid onto the cross-axis. The key words for it are stretch, flex-start, flex-end, center, and baseline. They can also be used with the key words of safe and unsafe.↩
                    </p>↩
                    <img src="Assets/Align_Items.JPG" alt="Align-items">↩
                <h3>Align-Content</h3>↩
                    <p>Aligns items if there is extra space inside the cross-axis through the use of the flex container’s lines. The key words for it are normal, flex-start, flex-end, center, space-between, space-around, space-evenly, and stretch. Like with Align-items, the key words safe and unsafe can be used with the other keywords.↩
                    </p>↩
```

```
78.                     <img src="Assets/Align_Content.JPG" alt="Align-content">↩
79.                 <h3>Gap</h3>↩
80.                     <p>This controls the gap between the items. It sets the
    minimum space between two items in the flex container.↩
81.                     </p>↩
82.                     <img src="Assets/Gap.JPG" alt="Gap">↩
83.             </div>↩
84.         </section>↩
85.         <section class="section4" id="section4">↩
86.             <h2>Flexbox Child Properties</h2>↩
87.             <div class="container">↩
88.                 <p>Unlike with flexbox’s parent properties, child properties in
    flexbox only affect items within the flexbox that have been specified by the web
    developer's code.↩
89.                 </p>↩
90.                 <h3>Order</h3>↩
91.                     <p>This controls the order that items are arranged within
    the flex container. The property values that are used for this property are the
    number that specifies the order of the item, initial which sets the property to
    the default, and inherit which causes the item to have the same property as its
    parent element.↩
92.                     </p>↩
93.                     <img src="Assets/Order.JPG" alt="Order">↩
94.                 <h3>Flex-Grow</h3>↩
95.                     <p>This defines how items will grow, with different items
    able to have different growth rates. The property values that are used for this
    property are the number that determines the amount the item will grow in
    comparison to the other items in the flexbox, initial which sets the value of
    the property to the default value, and inherit which makes the item take the
    value from its parent element.↩
96.                     </p>↩
97.                     <img src="Assets/Flex_Grow.JPG" alt="Flex-grow">↩
98.                 <h3>Flex-Shrink</h3>↩
99.                     <p>Defines how an item is able to shrink. The property
    values that are used for this property are the number that determines the amount
    the item will shrink in comparison to the other items, initial which sets the
    value of the property to the default value, and inherit which makes the item
    take the value from its parent.↩
100.                    </p>↩
101.                    <img src="Assets/Flex_Shrink.JPG" alt="Flex-Shrink">↩
102.                <h3>Flex-Basis</h3>↩
103.                    <p>Defines the size of an element within the flex container
    before what space remains is distributed. The property values that are used for
    this property are the number which sets the initial length of the item in either
    a length unit or a percent, auto which is the default value, initial which sets
    the value of the property to the default, and inherit which makes the item take
    the value from its parent.↩
104.                    </p>↩
105.                    <img src="Assets/Flex_Basis.JPG" alt="Flex-basis">↩
106.                <h3>Flex</h3>↩
107.                    <p>This combines flex-grow, flex-shrink, and flex-basis into
    one. The property values that are used are the same ones that are used for flex-
    grow, flex-shrink, and flex-basis.</p>↩
108.                    <img src="Assets/Flex.JPG" alt="Flex">↩
109.                <h3>Align-Self</h3>↩
110.                    <p>Allows for the default alignment to be overhauled for a
    specific item in the flex container. The property values that are used for this
    are auto, stretch, center, flex-start, flex-end, baseline, initial, and
    inherit.↩
111.                    </p>↩
112.                    <img src="Assets/Align_self.JPG" alt="Align-self">↩
113.            </div>↩
114.        </section>↩
115.    </main>↩
116.    <footer>↩
117.        <a href="#top">Back to Top</a>↩
```

```
118.         <h4>Colophon</h4>↩
119.         <h5>For the information used in this webpage, I used these websites for
     help getting the information.</h5>↩
120.             <a href="https://www.w3schools.com/csS/css3_flexbox.asp">W3Schools
     Flexbox</a>↩
121.             <a href="https://css-tricks.com/snippets/css/a-guide-to-
     flexbox/">CSS Tricks</a>↩
122.             <a href="https://medium.com/@PraveenPoonia/flexbox-a-flexible-css-
     layout-cd3e9da0c8f2">Medium</a> <br>↩
123.             <a href="https://medium.com/@BennyOgidan/history-of-css-grid-and-
     css-flexbox-658ae6cfe6d2">Medium</a>↩
124.             <a href="https://annairish.github.io/historicizing/history">Flexbox
     History</a>↩
125.             <a
     href="https://www.w3schools.com/cssref/css3_pr_order.php">W3Schools Order</a>
     <br>↩
126.             <a href="https://www.w3schools.com/cssref/css3_pr_flex-
     direction.php">W3Schools Flex-direction</a>↩
127.             <a href="https://www.w3schools.com/cssref/css3_pr_flex-shrink.php
     ">W3Schools Flex-Shrink</a>↩
128.             <a href="https://www.w3schools.com/cssref/css3_pr_flex.php ↩
129.             ">W3Schools Flex</a> <br>↩
130.             <a href="https://www.w3schools.com/cssref/css3_pr_flex-
     grow.php">W3Schools Flex-Grow</a>↩
131.             <a href="https://www.w3schools.com/cssref/css3_pr_flex-
     basis.php">W3SChools Flex-Basis</a>↩
132.             <a href="https://www.w3schools.com/cssref/css3_pr_flex-
     wrap.php">W3Schools Flex-Wrap</a> <br>↩
133.             <a href="https://www.w3schools.com/cssref/css3_pr_flex-
     flow.php">W3Schools Flex-Flow</a>↩
134.             <a href="https://www.w3schools.com/cssref/css3_pr_align-
     content.php">W3Schools Align-Content</a>↩
135.             <a href="https://www.w3schools.com/cssref/css3_pr_align-
     items.php">W3Schools Align-Items</a> <br>↩
136.             <a href="https://www.w3schools.com/cssref/css3_pr_align-
     self.php">W3Schools Align-Self</a>↩
137.             <a
     href="https://www.w3schools.com/cssref/pr_class_display.php">W3Schools
     Display</a>↩
138.             <a href="https://www.w3schools.com/cssref/css3_pr_gap.php">W3Schools
     Gap</a> <br>↩
139.             <a href="https://www.w3schools.com/cssref/css3_pr_justify-
     content.php">W3Schools Justify-Content</a>↩
140.         <h5>For the images used in this webpage, I used these websites.</h5> ↩
141.             <a href="https://www.w3schools.com/csS/css3_flexbox.asp">W3Schools
     Flexbox</a>↩
142.             <a
     href="https://www.w3schools.com/cssref/css3_pr_order.php">W3Schools Order</a>↩
143.             <a href="https://www.w3schools.com/cssref/css3_pr_flex-
     direction.php">W3Schools Flex-direction</a> <br>↩
144.             <a href="https://www.w3schools.com/cssref/css3_pr_flex-shrink.php
     ">W3Schools Flex-Shrink</a>↩
145.             <a href="https://www.w3schools.com/cssref/css3_pr_flex.php ↩
146.             ">W3Schools Flex</a>↩
147.             <a href="https://www.w3schools.com/cssref/css3_pr_flex-
     grow.php">W3Schools Flex-Grow</a> <br>↩
148.             <a href="https://www.w3schools.com/cssref/css3_pr_flex-
     basis.php">W3SChools Flex-Basis</a>↩
149.             <a href="https://www.w3schools.com/cssref/css3_pr_flex-
     wrap.php">W3Schools Flex-Wrap</a>↩
150.             <a href="https://www.w3schools.com/cssref/css3_pr_flex-
     flow.php">W3Schools Flex-Flow</a> <br>↩
151.             <a href="https://www.w3schools.com/cssref/css3_pr_align-
     content.php">W3Schools Align-Content</a>↩
152.             <a href="https://www.w3schools.com/cssref/css3_pr_align-
     items.php">W3Schools Align-Items</a>↩
```

```
153.             <a href="https://www.w3schools.com/cssref/css3_pr_align-
     self.php">W3Schools Align-Self</a> <br>↩
154.             <a
     href="https://www.w3schools.com/cssref/pr_class_display.php">W3Schools
     Display</a>↩
155.             <a href="https://www.w3schools.com/cssref/css3_pr_gap.php">W3Schools
     Gap</a>↩
156.             <a href="https://www.w3schools.com/cssref/css3_pr_justify-
     content.php">W3Schools Justify-Content</a> <br>↩
157.             <a href="https://annairish.github.io/historicizing/history">Flexbox
     History</a>↩
158.             <a
     href="https://www.w3schools.com/browsers/browsers_firefox.asp">W3Schools
     Browsers</a>↩
159.             <a href="https://css-tricks.com/snippets/css/a-guide-to-
     flexbox/">CSS Tricks</a>↩
160.     </footer>↩
161.     <script src="JS_Template/script.js"></script>↩
162.     </div>↩
163.     </body>↩
164. </html>
```

Used the HTML parser.

Total execution time 12 milliseconds.

About this checker • Report an issue • Version: 25.3.6