Front End Technologies

CSS - Day 9

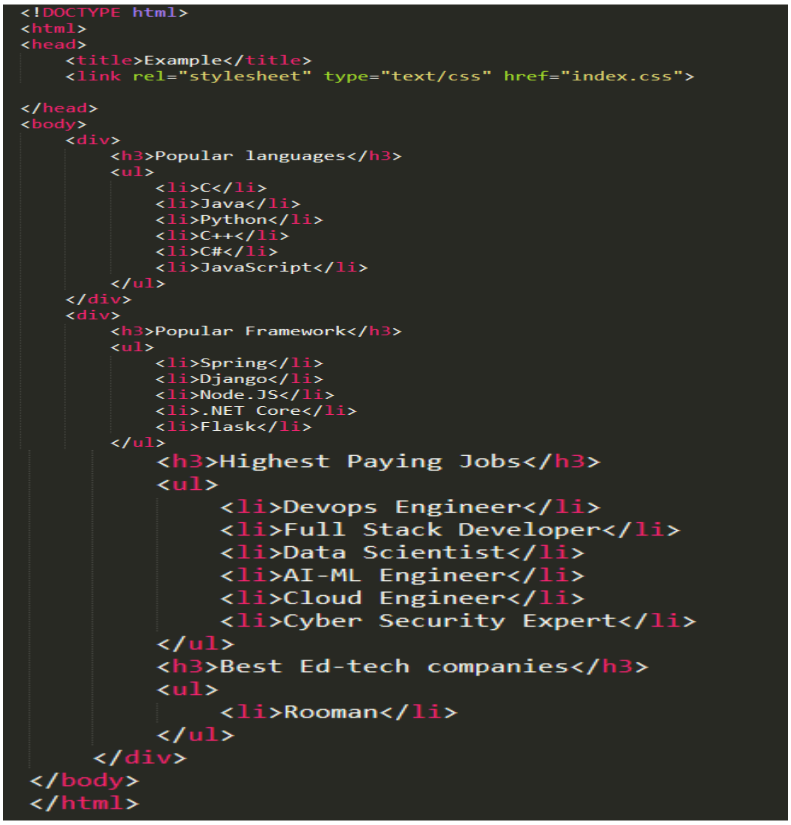
Agenda

- Pseudo classes continued

Combining first-child and last-child pseudo class:

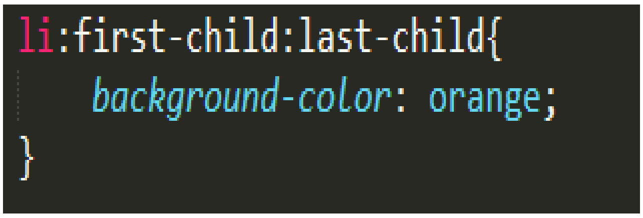
Example:

index.html



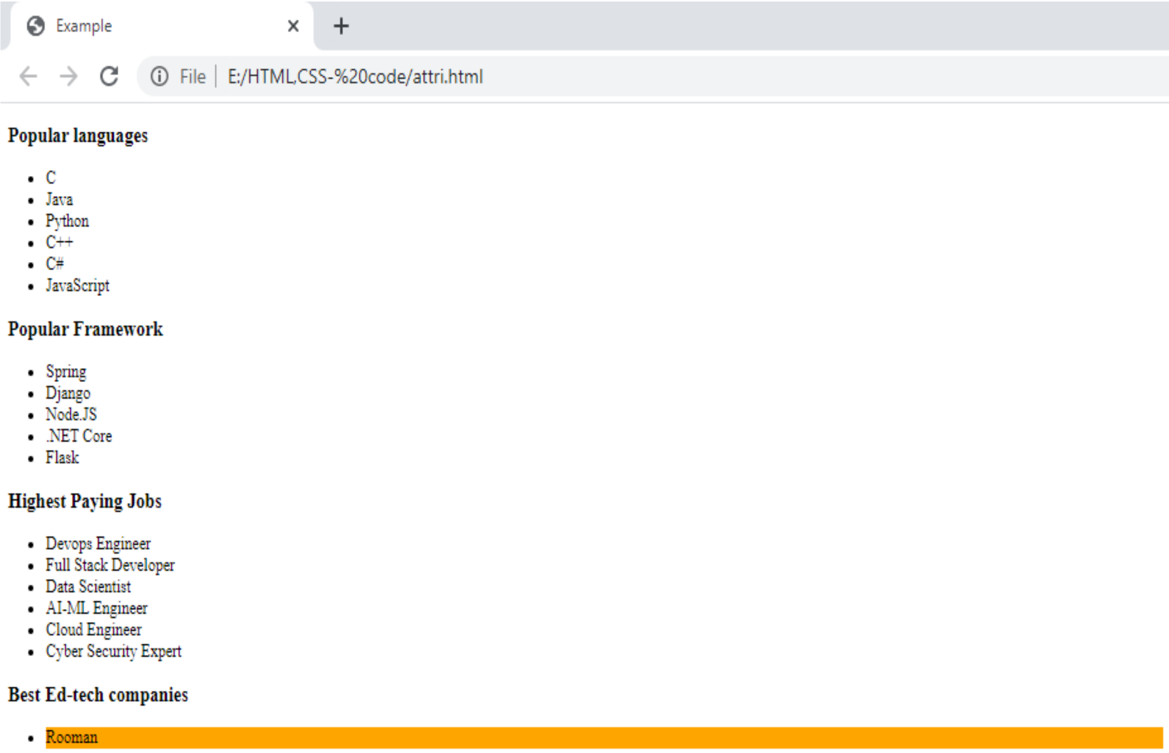
```
<!DOCTYPE html>
<html>
<head>
    <title>Example</title>
    <link rel="stylesheet" type="text/css" href="index.css">

</head>
<body>
    <div>
        <h3>Popular languages</h3>
        <ul>
            <li>C</li>
            <li>Java</li>
            <li>Python</li>
            <li>C++</li>
            <li>C#</li>
            <li>JavaScript</li>
        </ul>
    </div>
    <div>
        <h3>Popular Framework</h3>
        <ul>
            <li>Spring</li>
            <li>Django</li>
            <li>Node.JS</li>
            <li>.NET Core</li>
            <li>Flask</li>
        </ul>
        <h3>Highest Paying Jobs</h3>
        <ul>
            <li>Devops Engineer</li>
            <li>Full Stack Developer</li>
            <li>Data Scientist</li>
            <li>AI-ML Engineer</li>
            <li>Cloud Engineer</li>
            <li>Cyber Security Expert</li>
        </ul>
        <h3>Best Ed-tech companies</h3>
        <ul>
            <li>Rooman</li>
        </ul>
    </div>
</body>
</html>
```

index.css



```css
li:first-child:last-child{
    background-color: orange;
}
```

Output:



Popular languages

- C
- Java
- Python
- C++
- C#
- JavaScript

Popular Framework

- Spring
- Django
- Node.JS
- .NET Core
- Flask

Highest Paying Jobs

- Devops Engineer
- Full Stack Developer
- Data Scientist
- AI-ML Engineer
- Cloud Engineer
- Cyber Security Expert

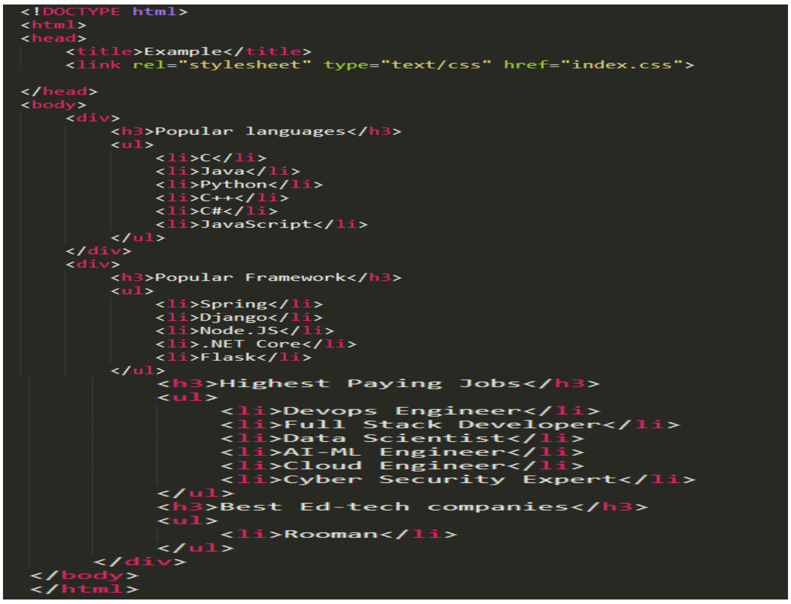
Best Ed-tech companies

- Rooman

In this example, only-child is selected using first-child and last-child pseudo class because only child is nothing but first and last child.

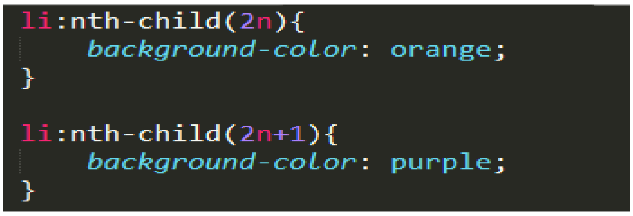
Example: Style the odd and even elements in the list using nth-child pseudo class

index.html



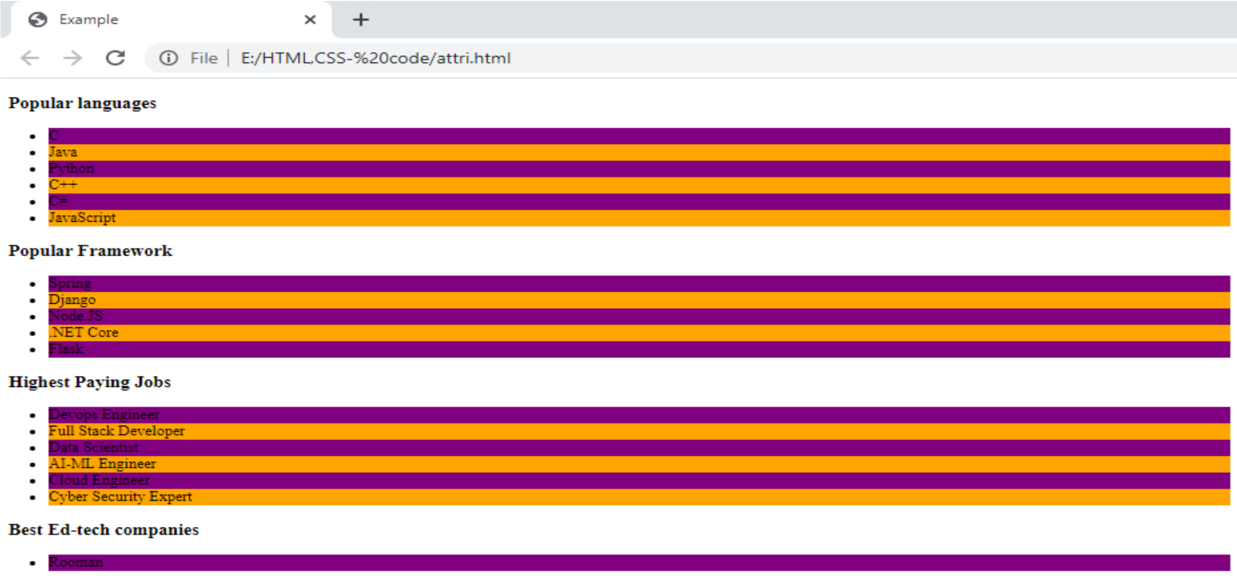
```html
<!DOCTYPE html>
<html>
<head>
    <title>Example</title>
    <link rel="stylesheet" type="text/css" href="index.css">

</head>
<body>
    <div>
        <h3>Popular languages</h3>
        <ul>
            <li>C</li>
            <li>Java</li>
            <li>Python</li>
            <li>C++</li>
            <li>C#</li>
            <li>JavaScript</li>
        </ul>
    </div>
    <div>
        <h3>Popular Framework</h3>
        <ul>
            <li>Spring</li>
            <li>Django</li>
            <li>Node.JS</li>
            <li>.NET Core</li>
            <li>Flask</li>
        </ul>
            <h3>Highest Paying Jobs</h3>
            <ul>
                <li>Devops Engineer</li>
                <li>Full Stack Developer</li>
                <li>Data Scientist</li>
                <li>AI-ML Engineer</li>
                <li>Cloud Engineer</li>
                <li>Cyber Security Expert</li>
            </ul>
            <h3>Best Ed-tech companies</h3>
            <ul>
                <li>Rooman</li>
            </ul>
      </div>
 </body>
 </html>
```

index.css



```css
li:nth-child(2n){
    background-color: orange;
}

li:nth-child(2n+1){
    background-color: purple;
}
```

Output:



Popular languages

- C
- Java
- Python
- C++
- C#
- JavaScript

Popular Framework

- Spring
- Django
- Node.JS
- .NET Core
- Flask

Highest Paying Jobs

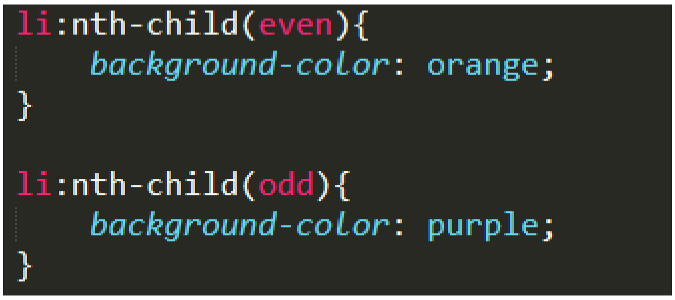
- Devops Engineer
- Full Stack Developer
- Data Scientist
- AI-ML Engineer
- Cloud Engineer
- Cyber Security Expert

Best Ed-tech companies

- Rooman

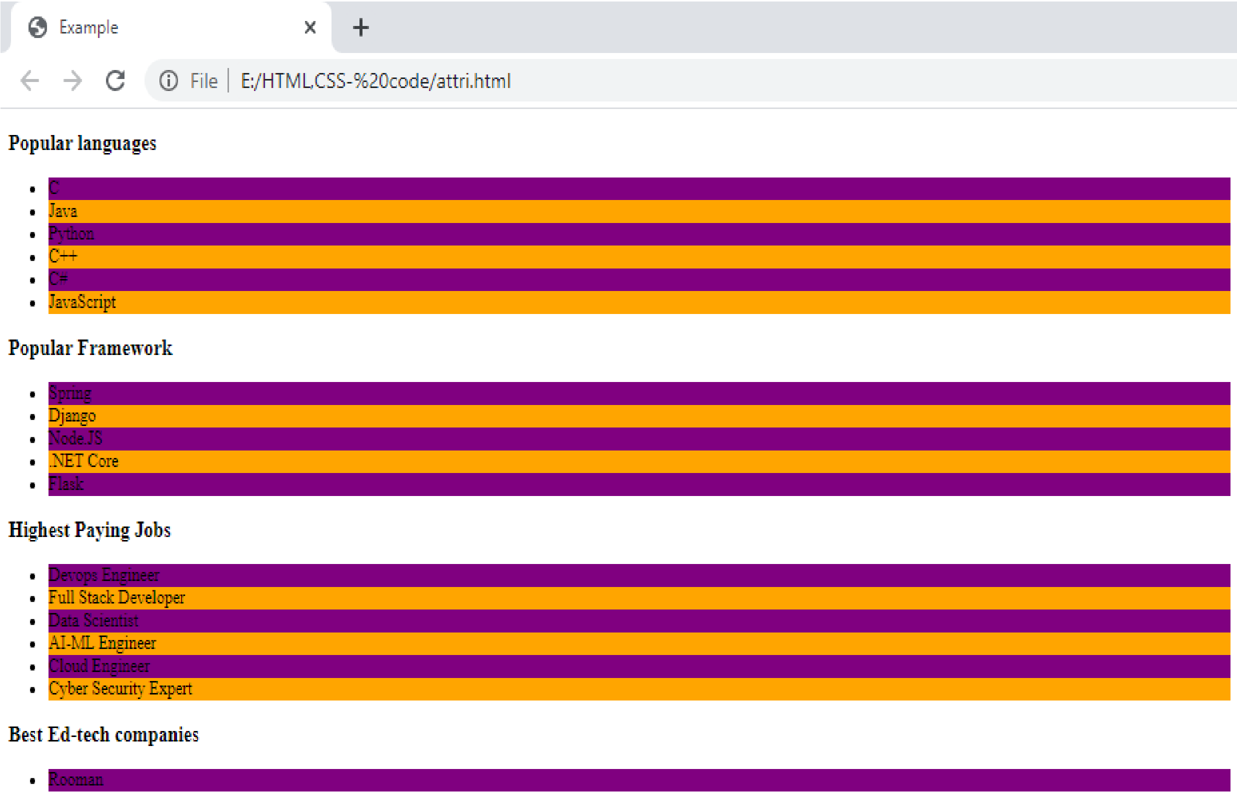
Here, nth-child(2n) will select all list items with even index numbers(2, 4, 6..etc) and nth-child(2n + 1) will select all list items with odd index numbers(1, 3, 5..etc). Instead of saying 2n and 2n + 1 in general you can say as even and odd as shown in below example.

index.css



```
li:nth-child(even){
    background-color: orange;
}

li:nth-child(odd){
    background-color: purple;
}
```

Output:



Popular languages

- C
- Java
- Python
- C++
- C#
- JavaScript

Popular Framework

- Spring
- Django
- Node.JS
- .NET Core
- Flask

Highest Paying Jobs

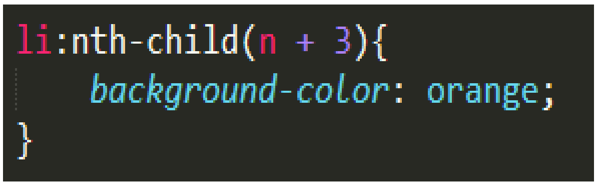
- Devops Engineer
- Full Stack Developer
- Data Scientist
- AI-ML Engineer
- Cloud Engineer
- Cyber Security Expert

Best Ed-tech companies

- Rooman

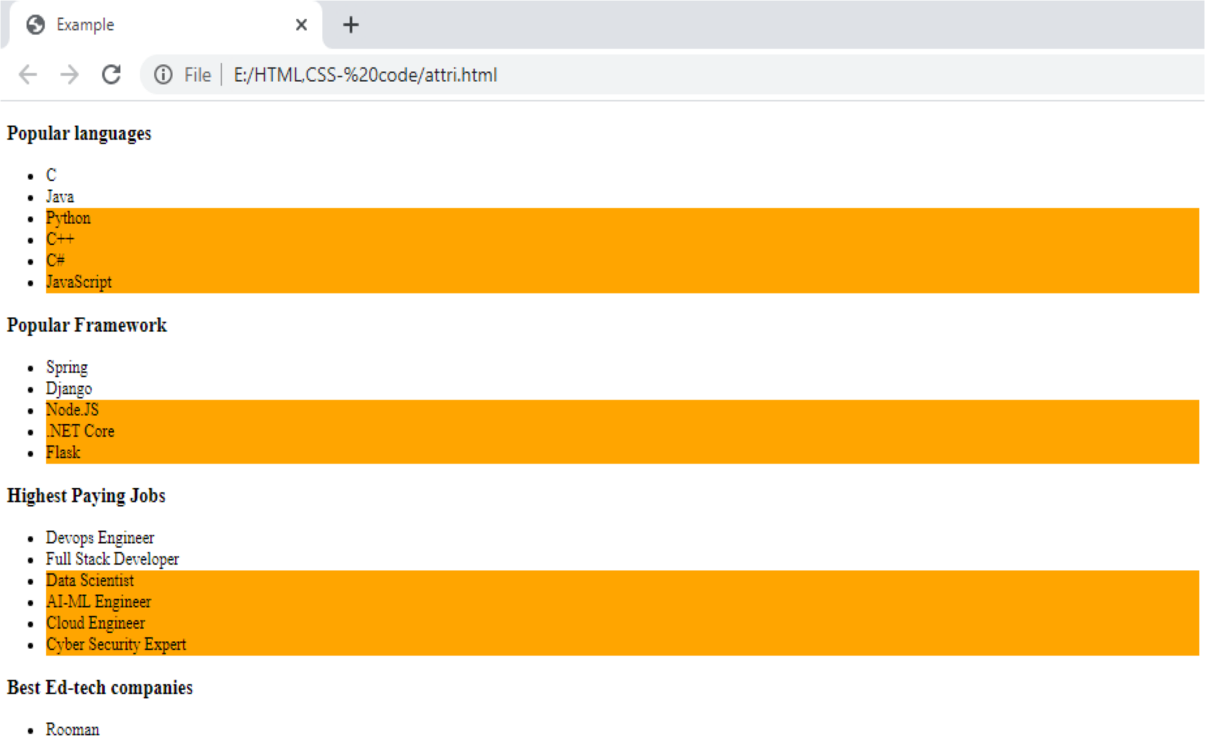
Example: Select range of values using nth-child

Index.css



```css
li:nth-child(n + 3){
    background-color: orange;
}
```

Output:



Popular languages

- C
- Java
- Python
- C++
- C#
- JavaScript

Popular Framework

- Spring
- Django
- Node.JS
- .NET Core
- Flask

Highest Paying Jobs

- Devops Engineer
- Full Stack Developer
- Data Scientist
- AI-ML Engineer
- Cloud Engineer
- Cyber Security Expert

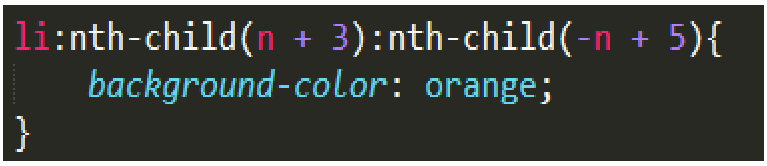
Best Ed-tech companies

- Rooman

In this example, (n + 3) selects every row starting from 3. If you want to select every row starting from 5 means you can write as (n + 5).

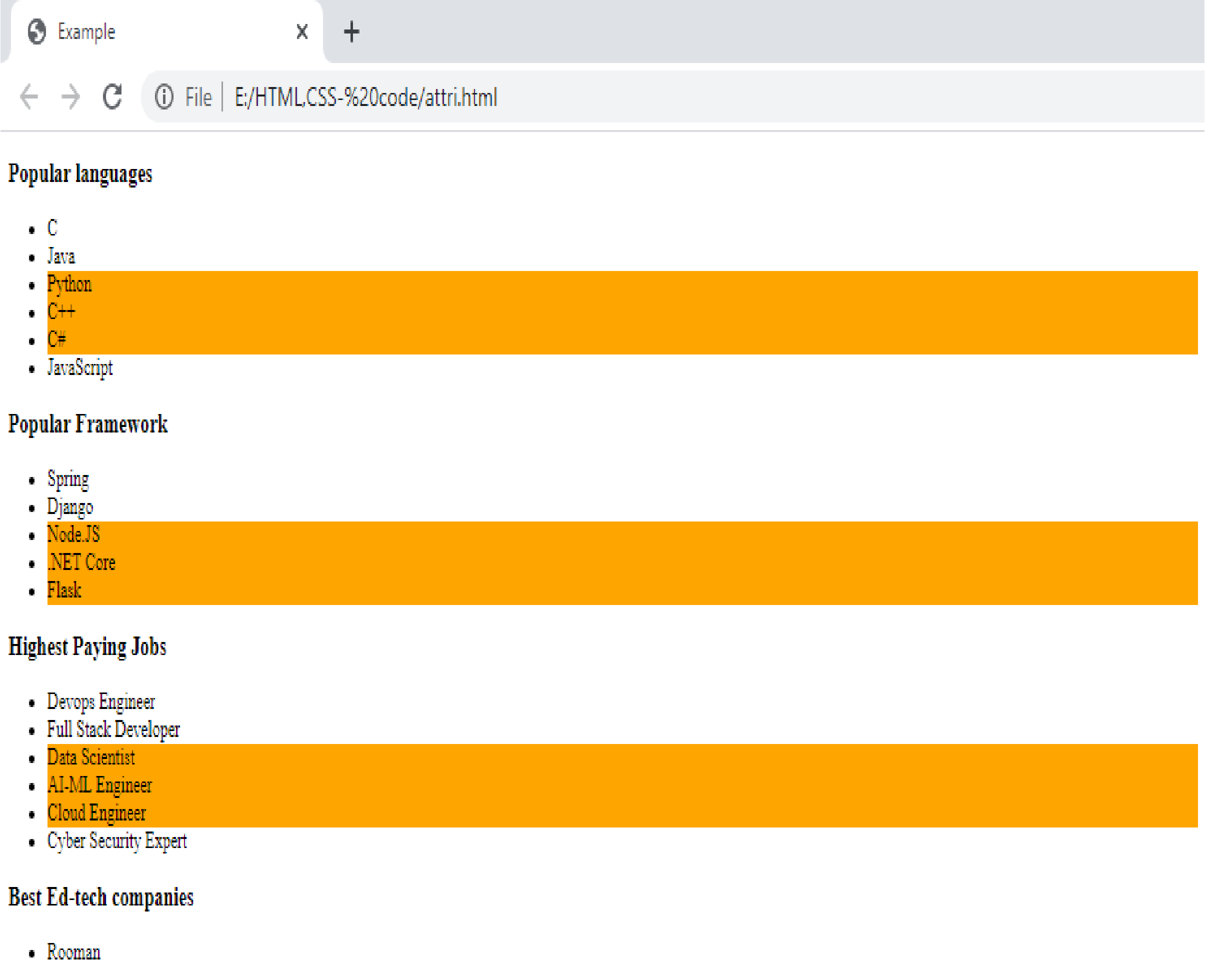
Example: Selecting the values from 3rd index to 5th index.

index.css



```css
li:nth-child(n + 3):nth-child(-n + 5){
    background-color: orange;
}
```

Output:



Popular languages

- C
- Java
- Python
- C++
- C#
- JavaScript

Popular Framework

- Spring
- Django
- Node.JS
- .NET Core
- Flask

Highest Paying Jobs

- Devops Engineer
- Full Stack Developer
- Data Scientist
- AI-ML Engineer
- Cloud Engineer
- Cyber Security Expert

Best Ed-tech companies

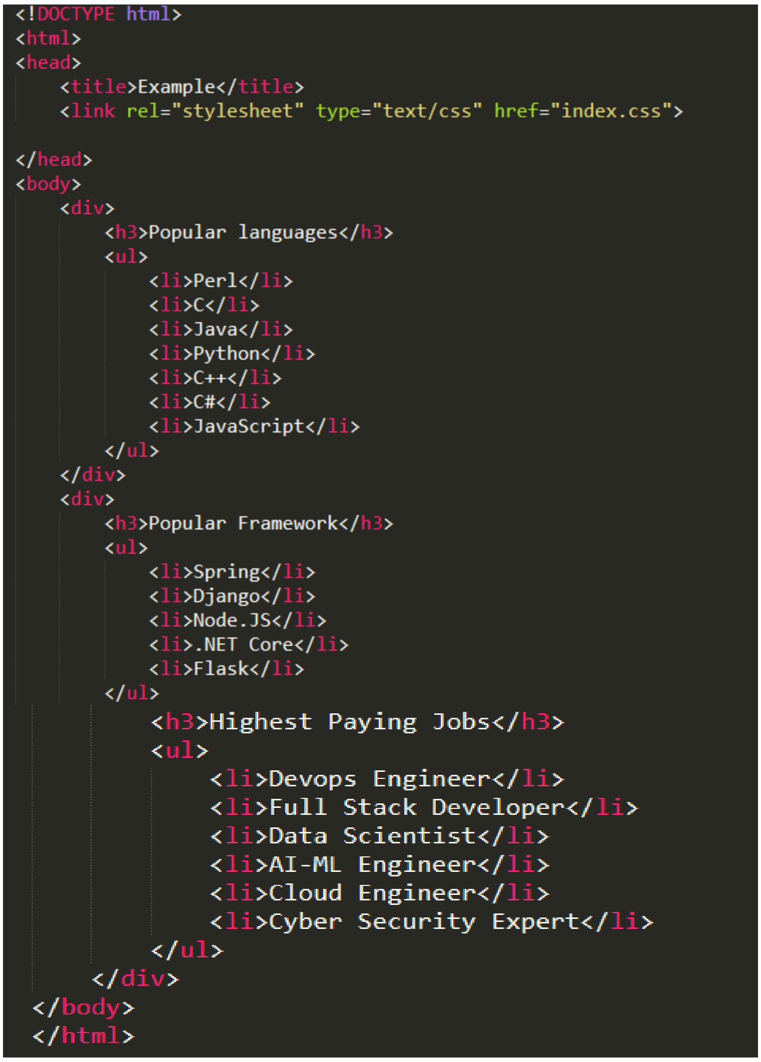
- Rooman

In this example, we have applied 2 nth-child selectors to one-group of siblings.

Only-of-type: only-of-type selector in CSS represents only those element that has no siblings of the given type.

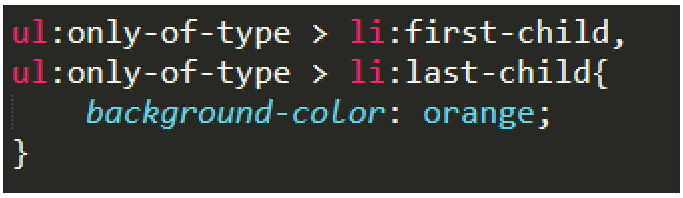
Example:

index.html



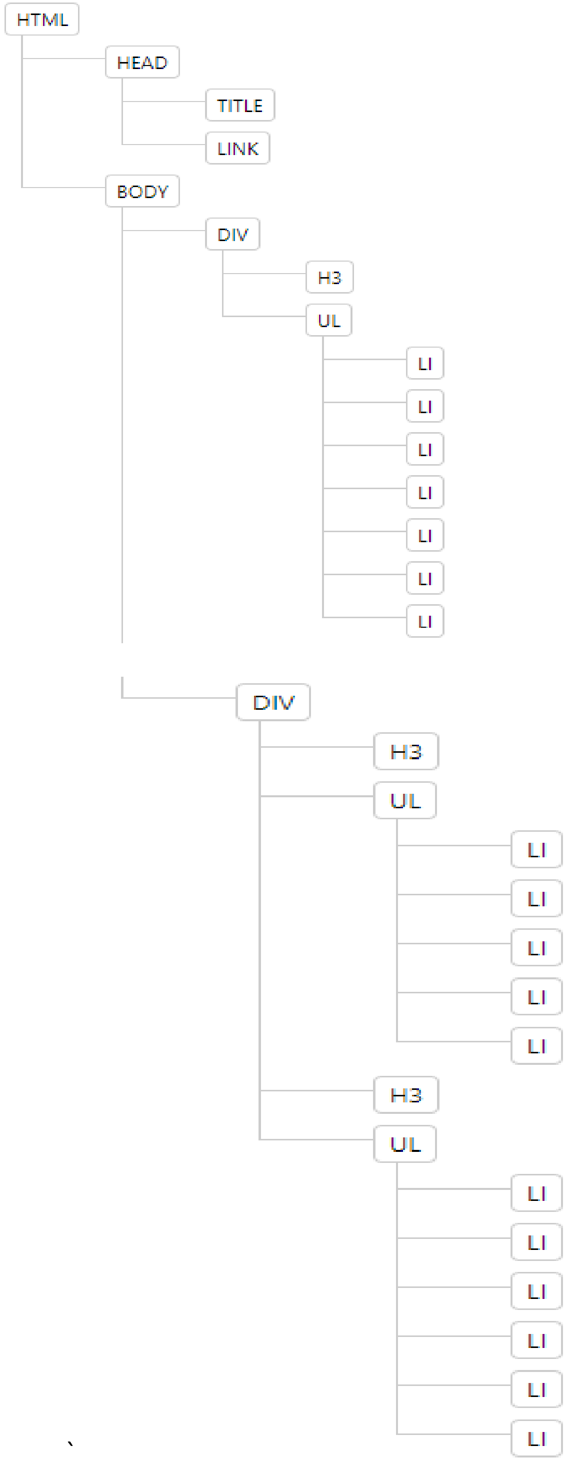
```html
<!DOCTYPE html>
<html>
<head>
    <title>Example</title>
    <link rel="stylesheet" type="text/css" href="index.css">

</head>
<body>
    <div>
        <h3>Popular languages</h3>
        <ul>
            <li>Perl</li>
            <li>C</li>
            <li>Java</li>
            <li>Python</li>
            <li>C++</li>
            <li>C#</li>
            <li>JavaScript</li>
        </ul>
    </div>
    <div>
        <h3>Popular Framework</h3>
        <ul>
            <li>Spring</li>
            <li>Django</li>
            <li>Node.JS</li>
            <li>.NET Core</li>
            <li>Flask</li>
        </ul>
        <h3>Highest Paying Jobs</h3>
        <ul>
            <li>Devops Engineer</li>
            <li>Full Stack Developer</li>
            <li>Data Scientist</li>
            <li>AI-ML Engineer</li>
            <li>Cloud Engineer</li>
            <li>Cyber Security Expert</li>
        </ul>
    </div>
</body>
</html>
```

Index.css



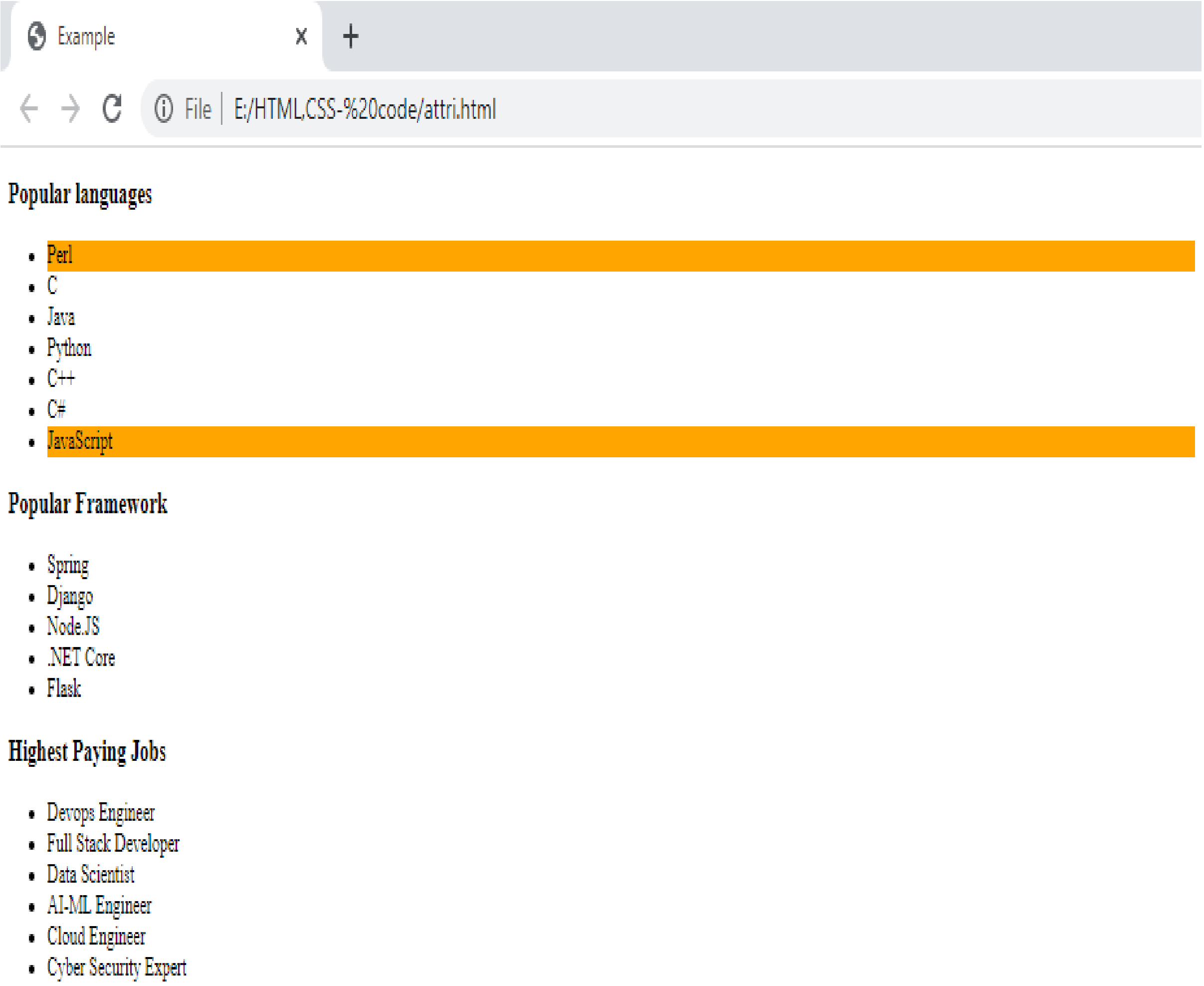
```css
ul:only-of-type > li:first-child,
ul:only-of-type > li:last-child{
    background-color: orange;
}
```

Tree Representation:



`

Output:



Popular languages

- Perl
- C
- Java
- Python
- C++
- C#
- JavaScript

Popular Framework

- Spring
- Django
- Node.JS
- .NET Core
- Flask

Highest Paying Jobs

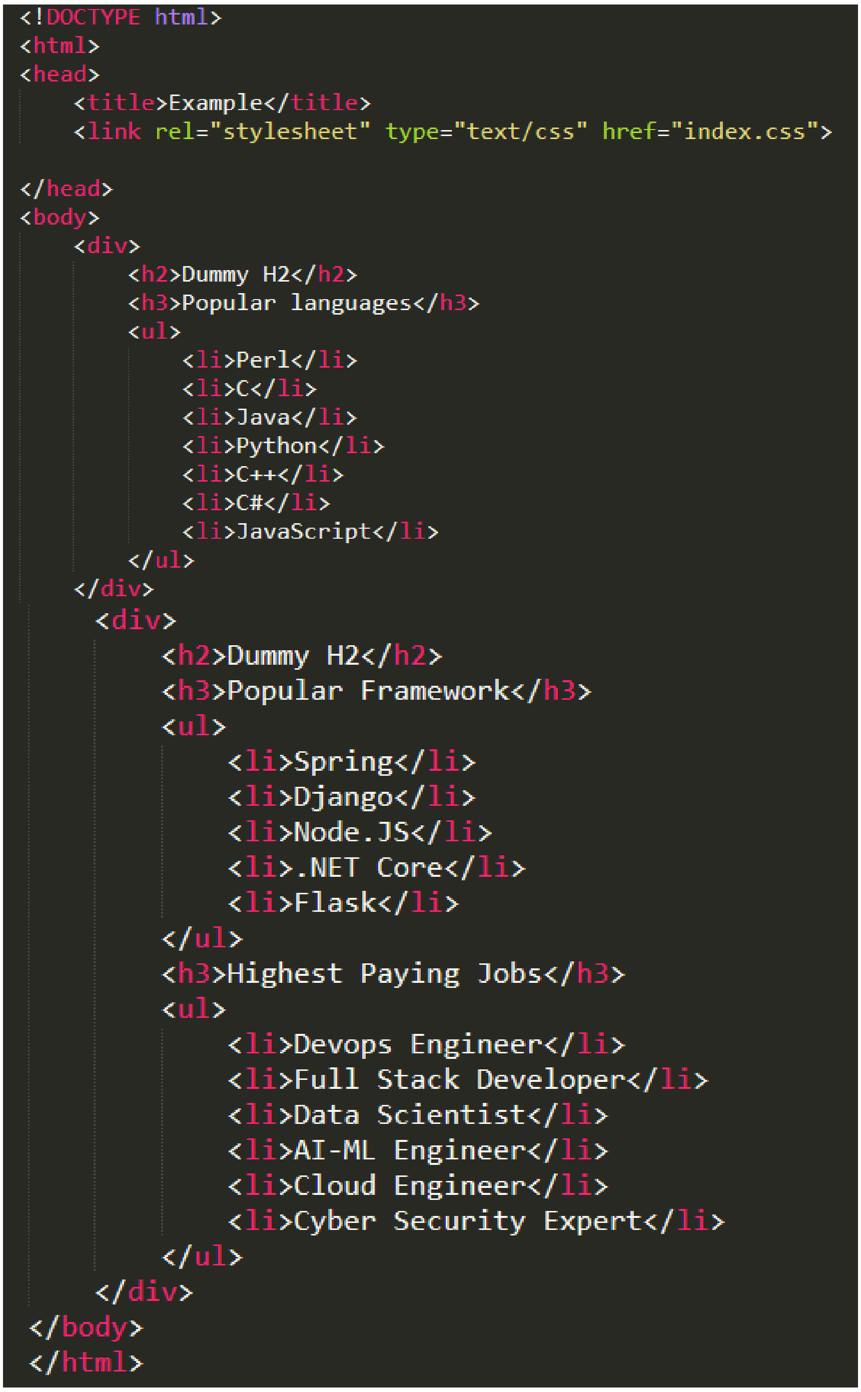
- Devops Engineer
- Full Stack Developer
- Data Scientist
- AI-ML Engineer
- Cloud Engineer
- Cyber Security Expert

In this example, styling is applied to first and last li present inside first ul. Only-of-type is used to select first ul, first-child and last-child is used to select first and last li inside ul.

first-of-type: first-of-type pseudo-class selects an element that is the first element of its type in the list of children of its parent. Let’s understand first-of-type pseudo class with an example.

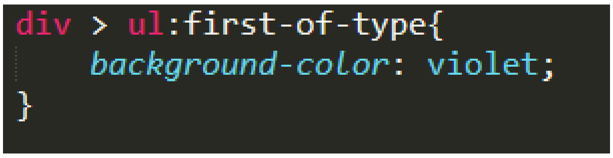
Example:

index.html



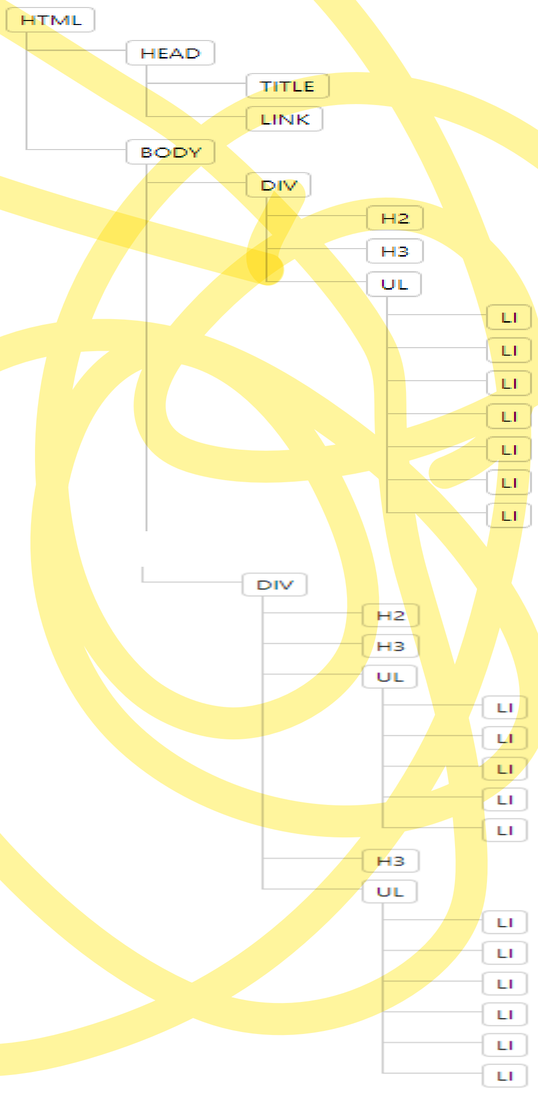
```html
<!DOCTYPE html>
<html>
<head>
    <title>Example</title>
    <link rel="stylesheet" type="text/css" href="index.css">

</head>
<body>
    <div>
        <h2>Dummy H2</h2>
        <h3>Popular languages</h3>
        <ul>
            <li>Perl</li>
            <li>C</li>
            <li>Java</li>
            <li>Python</li>
            <li>C++</li>
            <li>C#</li>
            <li>JavaScript</li>
        </ul>
    </div>
     <div>
         <h2>Dummy H2</h2>
         <h3>Popular Framework</h3>
         <ul>
             <li>Spring</li>
             <li>Django</li>
             <li>Node.JS</li>
             <li>.NET Core</li>
             <li>Flask</li>
         </ul>
         <h3>Highest Paying Jobs</h3>
         <ul>
             <li>Devops Engineer</li>
             <li>Full Stack Developer</li>
             <li>Data Scientist</li>
             <li>AI-ML Engineer</li>
             <li>Cloud Engineer</li>
             <li>Cyber Security Expert</li>
         </ul>
     </div>
 </body>
 </html>
```

index.css

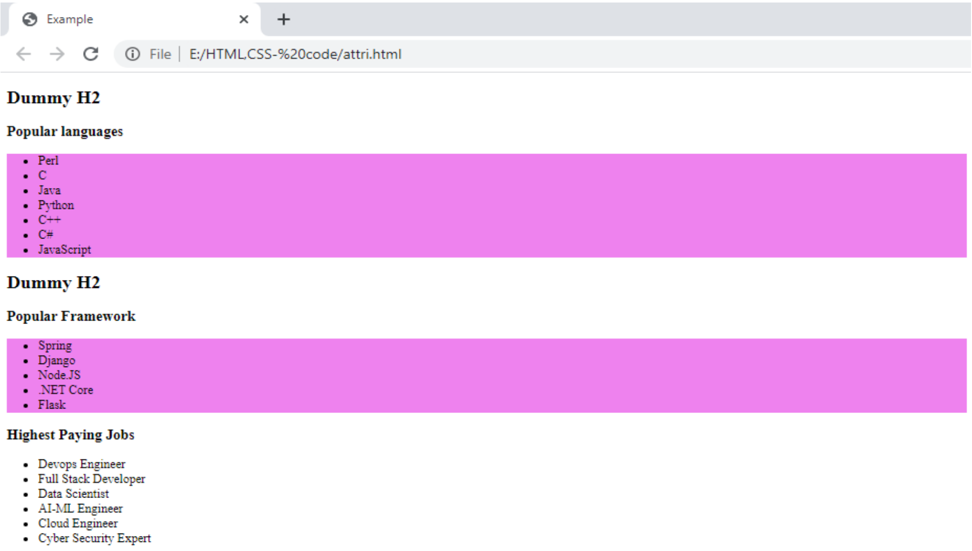


```css
div > ul:first-of-type{
    background-color: violet;
}
```

Tree representation:



Output:



Dummy H2

Popular languages

- Perl
- C
- Java
- Python
- C++
- C#
- JavaScript

Dummy H2

Popular Framework

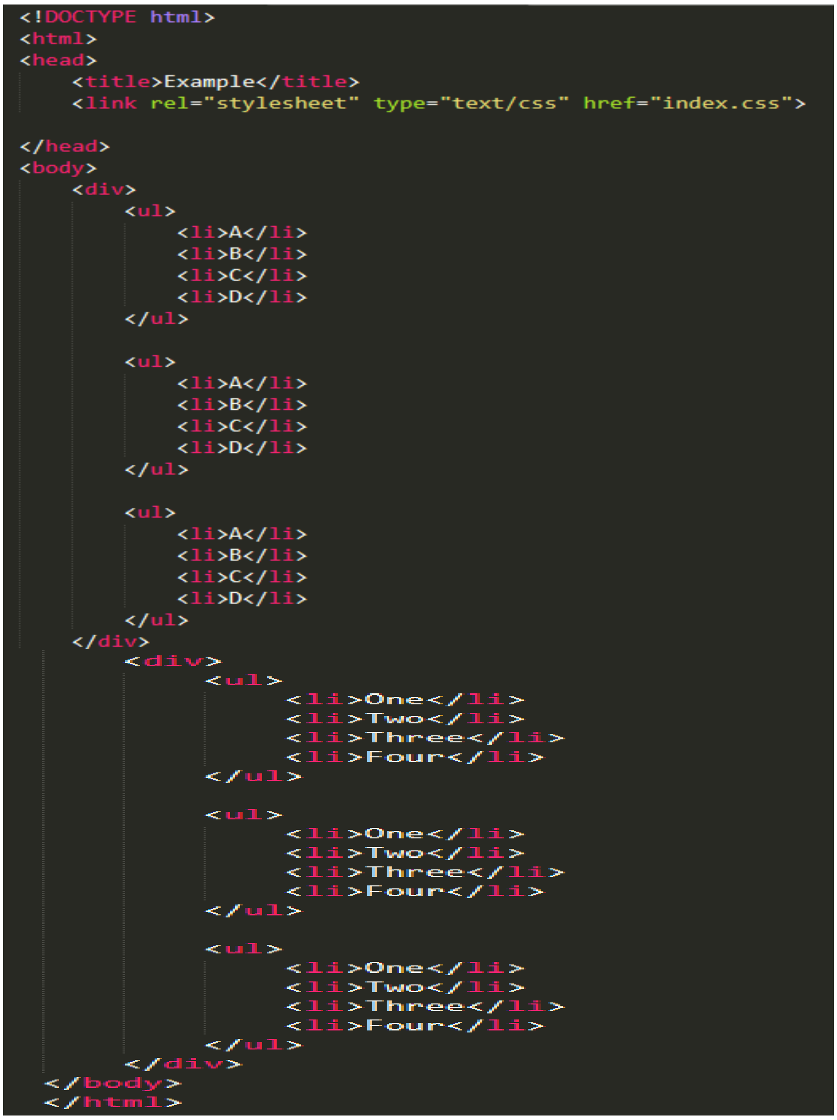
- Spring
- Django
- Node.JS
- .NET Core
- Flask

Highest Paying Jobs

- Devops Engineer
- Full Stack Developer
- Data Scientist
- AI-ML Engineer
- Cloud Engineer
- Cyber Security Expert

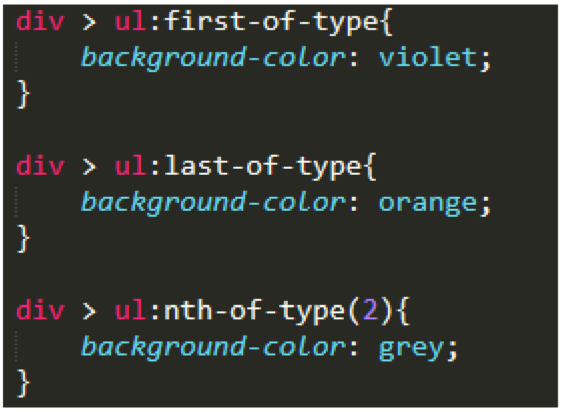
Example: Styling of list inside ul using firs-of-type, last-of-type and nth-of-type

Index.html



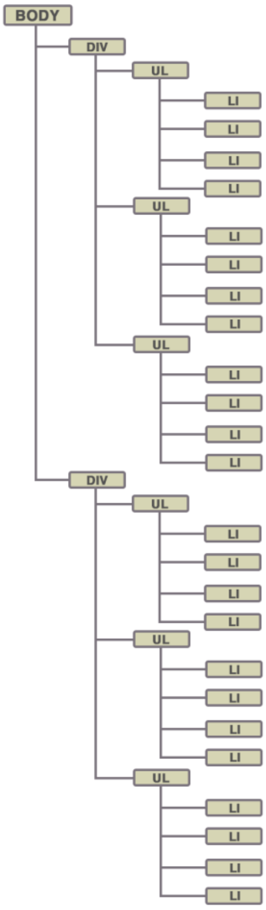
```
<!DOCTYPE html>
<html>
<head>
    <title>Example</title>
    <link rel="stylesheet" type="text/css" href="index.css">

</head>
<body>
    <div>
        <ul>
            <li>A</li>
            <li>B</li>
            <li>C</li>
            <li>D</li>
        </ul>

        <ul>
            <li>A</li>
            <li>B</li>
            <li>C</li>
            <li>D</li>
        </ul>

        <ul>
            <li>A</li>
            <li>B</li>
            <li>C</li>
            <li>D</li>
        </ul>
    </div>
        <div>
            <ul>
                <li>One</li>
                <li>Two</li>
                <li>Three</li>
                <li>Four</li>
            </ul>

            <ul>
                <li>One</li>
                <li>Two</li>
                <li>Three</li>
                <li>Four</li>
            </ul>

            <ul>
                <li>One</li>
                <li>Two</li>
                <li>Three</li>
                <li>Four</li>
            </ul>
        </div>
  </body>
  </html>
```

Index.css

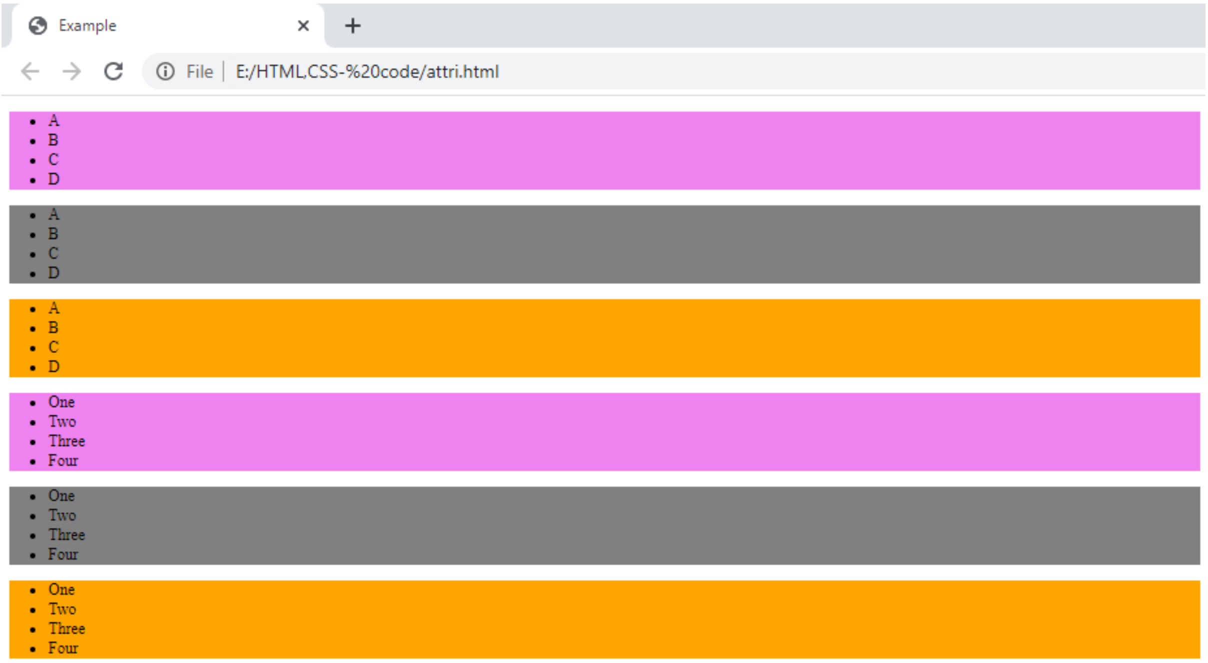


```
div > ul:first-of-type{
    background-color: violet;
}

div > ul:last-of-type{
    background-color: orange;
}

div > ul:nth-of-type(2){
    background-color: grey;
}
```

Tree representation:



Output:



- A
- B
- C
- D

- A
- B
- C
- D

- A
- B
- C
- D

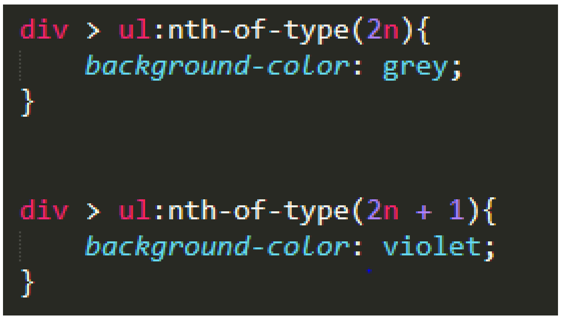
- One
- Two
- Three
- Four

- One
- Two
- Three
- Four

- One
- Two
- Three
- Four

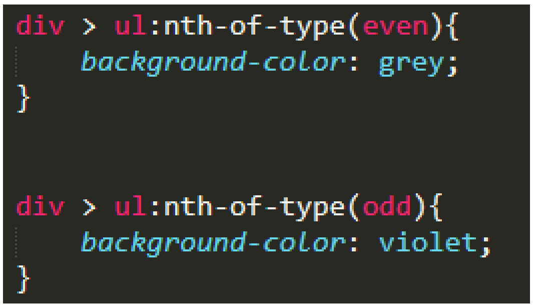
Example: Styling of even and odd ul inside the div. (refer index.html f above example)

index.css



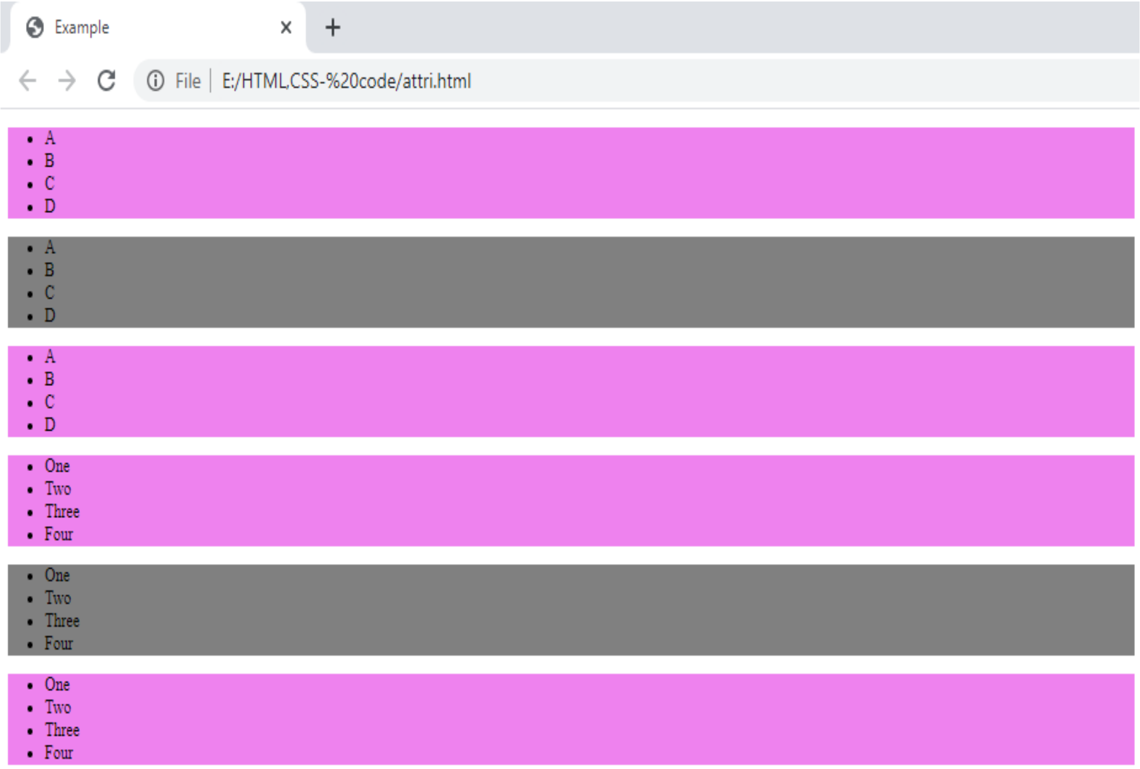
```css
div > ul:nth-of-type(2n){
    background-color: grey;
}


div > ul:nth-of-type(2n + 1){
    background-color: violet;
}
```

Or



```css
div > ul:nth-of-type(even){
    background-color: grey;
}


div > ul:nth-of-type(odd){
    background-color: violet;
}
```

Output:



```
• A
• B
• C
• D

• A
• B
• C
• D

• A
• B
• C
• D

• One
• Two
• Three
• Four

• One
• Two
• Three
• Four

• One
• Two
• Three
• Four
```