Overview   **Q&A**   **Files**   **Announcements**

CSS Tutorial: Float & Clear Explained | Web Development Tutorials #21

In this tutorial, we are going to see about different types of alignments available in CSS. We will start by making a new file called *tut21.html* and add the boilerplate to get the basic HTML code. Give the title as **Alignment** in the <title> tag.

The CSS **float** property specifies how an element should float. The CSS **clear** property specifies what elements can float beside the cleared element and on which side. The *float* property is used for positioning and formatting content, for example, let an image float left to the text in a container. The *float* property can have one of one of the following values-

- **Left-** The elements floats to the left of its container.
- **Right-** The elements floats to the right of its container.
- **None-** The element does not float (it will be displayed just where it occurs in the text). This is default.
- **Inherit-** The element inherits the float value of its parent.

Let us imagine that we are making a grocery store website and accordingly sell the things.

For the CSS section, we will make different IDs and classes to specify different properties to each item listed. Let us start by defining the classes

```
.container {
        width: 900px;
        border: 3px solid purple;
        background-color: rgb(250, 226, 205);
        margin: 33px auto;
    }

    .item {
        border: 3px solid grey;
        margin: 12px 3px;
        padding: 12px 3px;
        background: rgb(248, 238, 238);
    }
```

The *auto* property of margin allows to automatically adjust the margin equally on the both the ends. The result will be as follows as such-
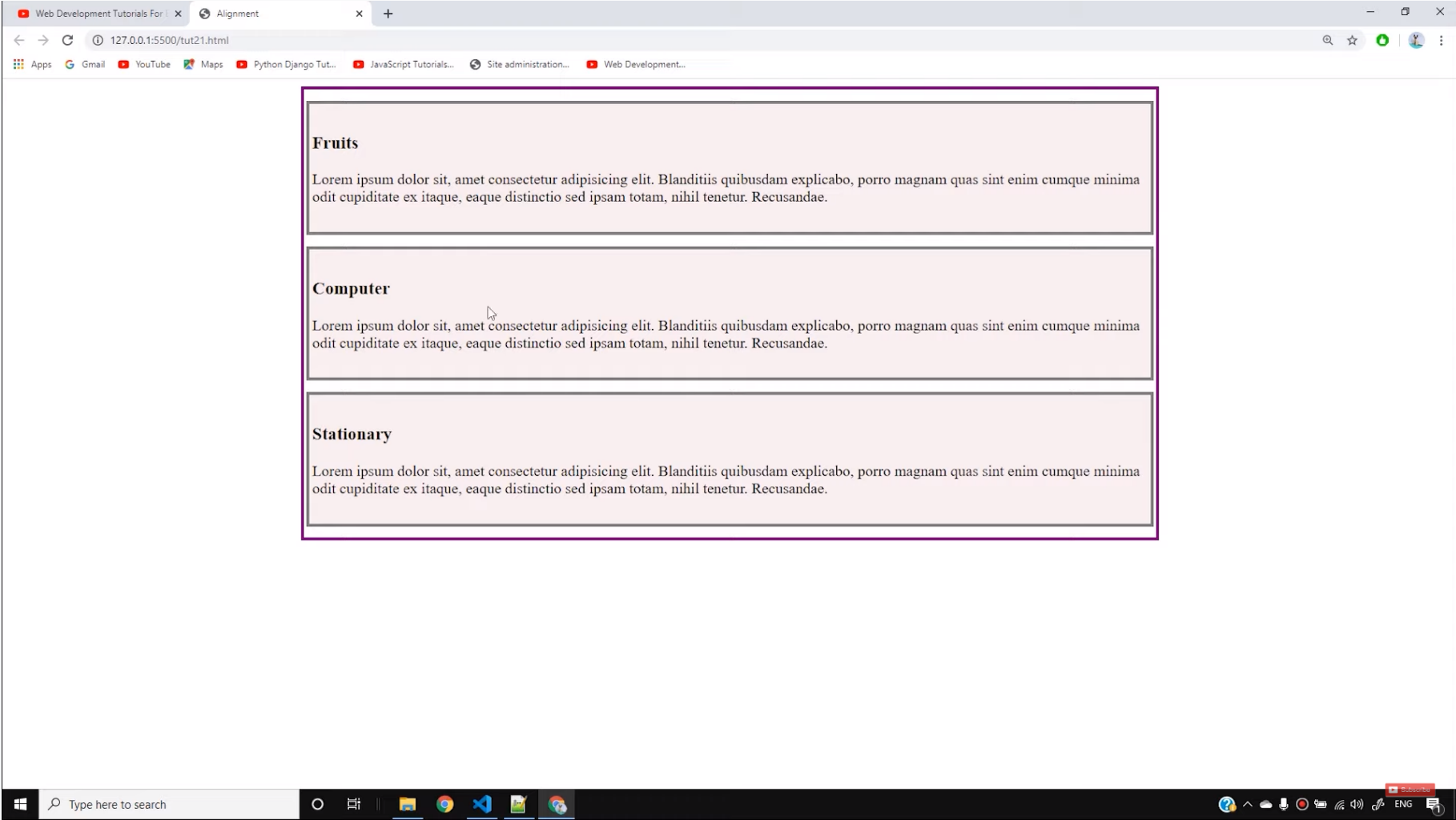

To float the elements, right or left we can target them by their IDs. Let us target all the elements as shown below-

```css
#fruit {
        float: right;
        width: 48%;
    }


    #computer {
        float: left;
        width: 48%;
    }


    #stationary {
        /* float: left; */
        clear: both;
        clear: left;
        width: 100%;
    }
```

Initially, if you set the width as *50%* for all three, then the result would be as follows-
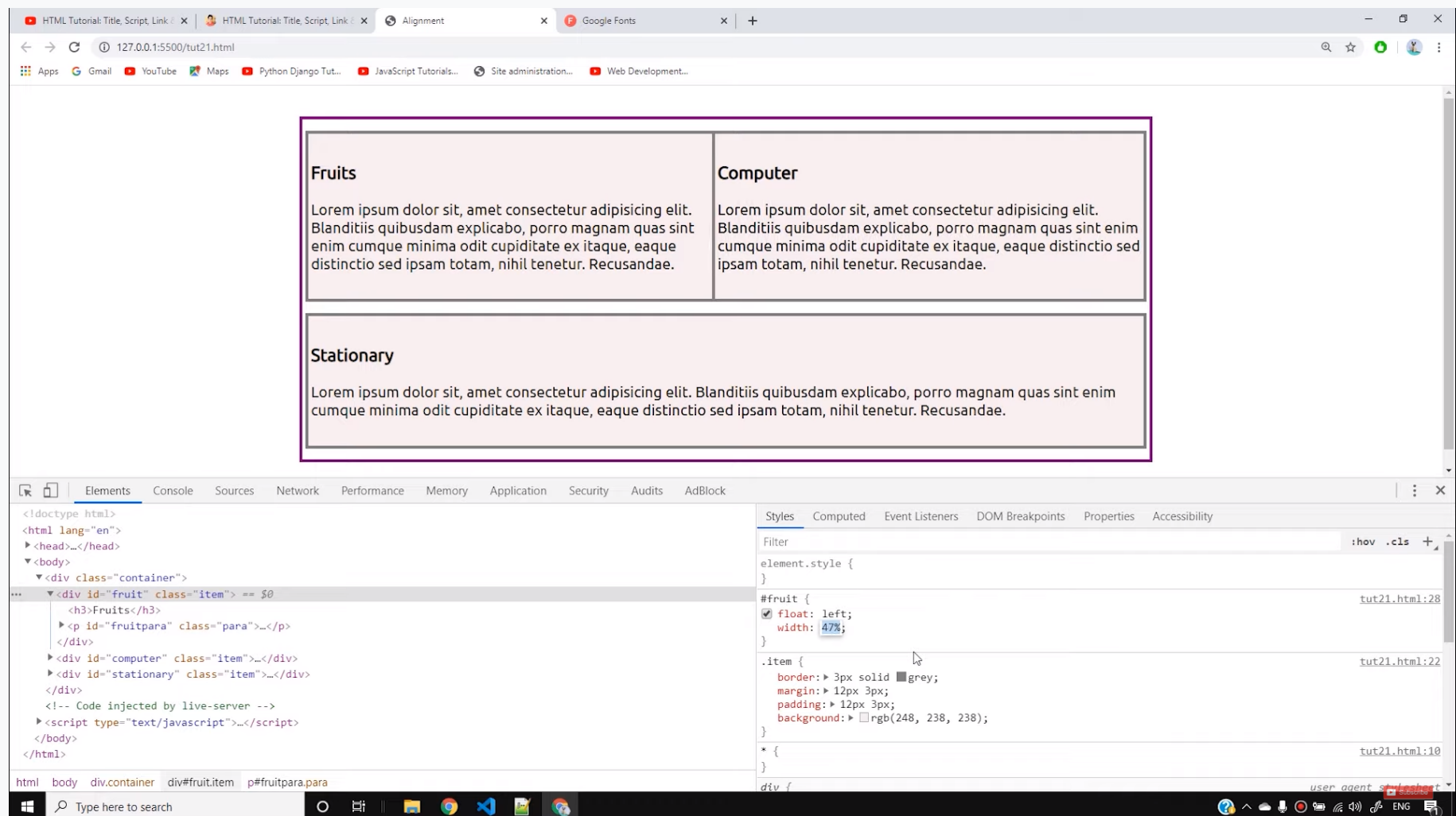
If we add some more texts to *fruit* and *computer* and remove the *float: left* option from *stationary* then we find that fruit and computer will float on the right side of the container and overlaps the stationary section as follows-
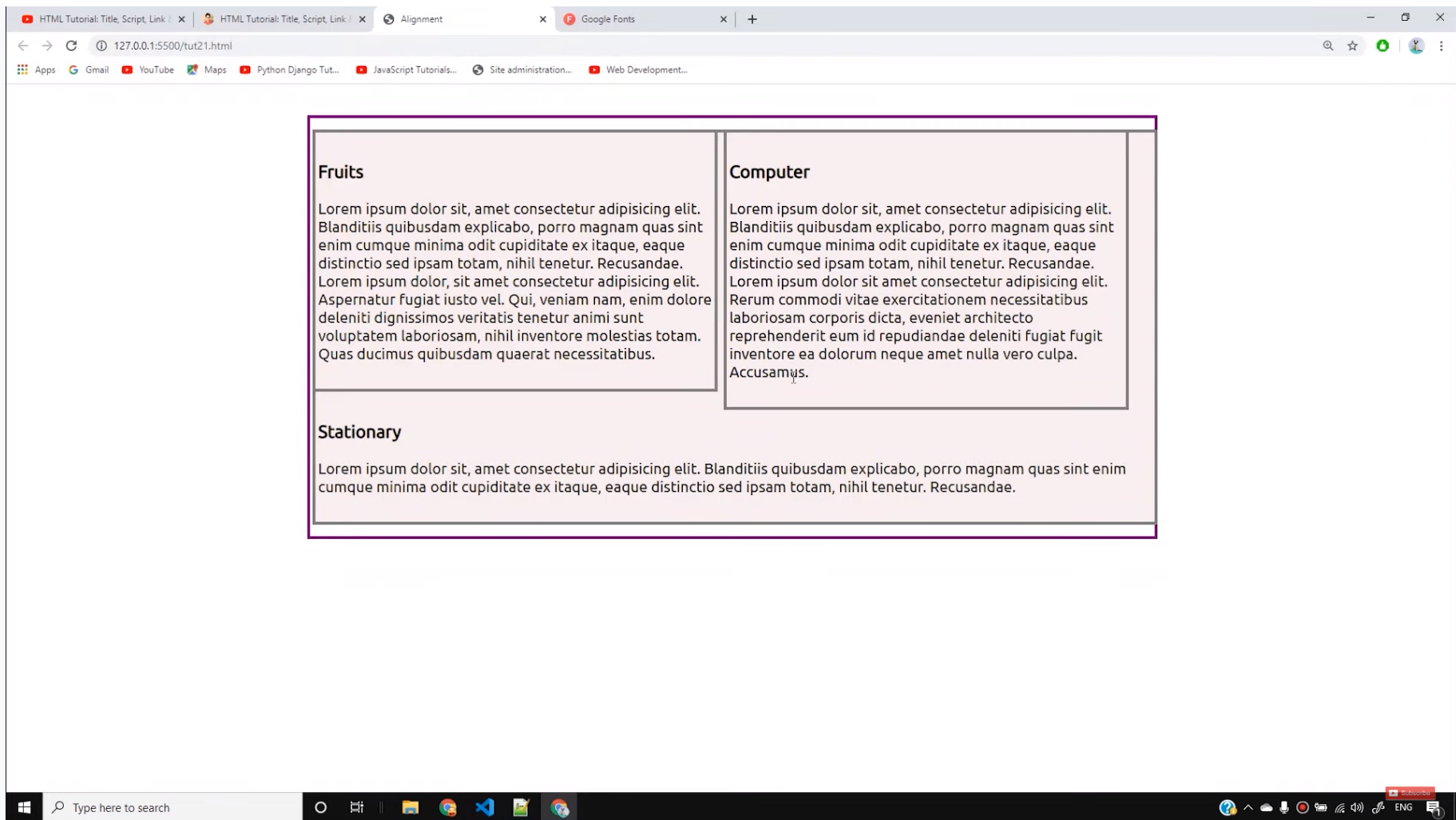
To avoid this, we use the property known as *clear.* If we write *clear: both,* then both the other elements will not overlap the *stationary section.*

For paragraphs, we have different alignments options like *right, left, center,* and *justify.* The right alignment will move the texts to the right, left alignment to the left side and so on.

```css
p, h3 {
        /* text-align: right;
    text-align: left;
    text-align: center; */
    text-align: justify;
    }
```

However, now we do not use the *float and clear* property much. Instead, we use the properties like *flexbox*. But then too the concepts of float and clear should be known to you. In the upcoming tutorials, we will see more different layouts like navigation bars etc. Till then stay tuned with the tutorials.

Code as described/written in the video

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Alignment</title>
    <link href="https://fonts.googleapis.com/css?family=Ubuntu&display=swap" rel="stylesheet">
    <style>
        * {
            box-sizing: border-box;
        }

        body {
            font-family: 'Ubuntu', sans-serif;
        }

        .container {
            width: 900px;
            border: 3px solid purple;
            background-color: rgb(250, 226, 205);
            margin: 33px auto;
        }
```

← Previous   Next →