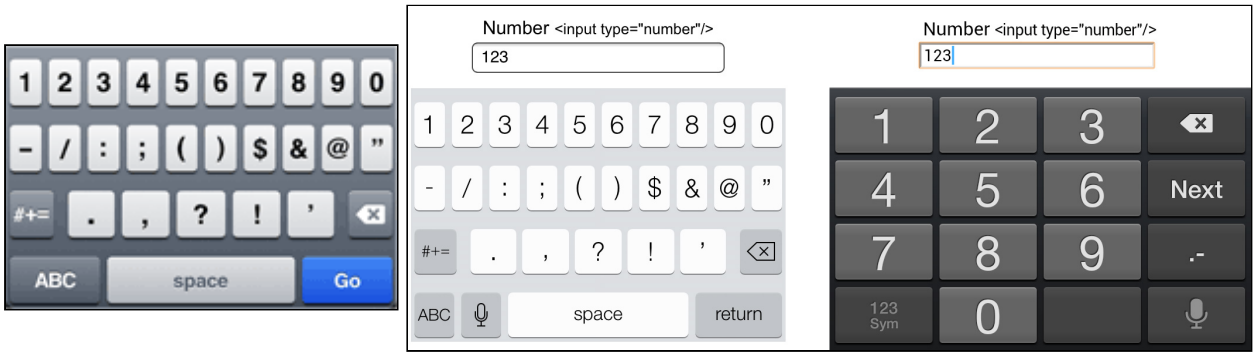
<input type="number">

This input field is useful for entering numerical values (integer or float), but not for entering zip codes. On desktop implementations and on some mobile implementations, it provides a user interface with small vertical arrows for incrementing/decrementing the current value, while on mobiles it will display a numeric keyboard.

For zip codes, a `<input type="text" pattern="......">`is preferable. See examples given in the the `pattern` attribute section of this course.

Example: `<input type="number" value="25" min="0" step="5" max="500"/>`

Screenshot examples taken with mobile devices (left: IOS, right: Android)



Examples on desktop (the width will be adjusted depending on the `min` and `max` attributes):

Quantity (between 1 and 5):

Quantity (between 1 and 5): 3 Soumettre

TYPICAL USAGE

```
<input type="number" value="25" min="0" step="5" max="500"/>
```

This field accepts specific attributes `max`, `min`, `step`, `value`(default displayed value).

This input type is very interesting as it provides default validation behaviors:

- If the value entered using a keyboard is not a valid number, or is not in the range defined by the `min` and `max`attributes, the field is *invalid* and gets the pseudo CSS class `:invalid`.
- If the the difference between the `value` you enter and`min` is a multiple of `step`, then it gets the CSS pseudo class `:valid`, otherwise it will be *invalid*. Example: if `min=1` and `step=5`, the field will be valid with `value=1, 6, 11, 16` etc. if `min=0`, with `value=0, 5, 10, 15` etc.

> **WARNING 1**: **Using a `step` attribute with an integer value will make the arrows increment/decrement the current value with the`step` value, and make the input field valid only when the difference between the `value` you enter and `min` is a multiple of `step`.**

> **WARNING 2: by default, omitting the `step`attribute is equivalent to `step="1"`, so for entering float values, it is necessary to use`step="any"` or step equal to a floating point value such as `step="0.1"`.**
>
> **With `step="any"`, floating point values are valid, but vertical arrows will increment/decrement the value by one. If`step="0.1"`, arrows will increment/decrement by `0.1`, etc.**

Online example at JS Bin: (try changing the attribute values, use`step="any"` and try float values, etc).

Or, do it here in your browser (Manually enter a value that is not in the range, or not a multiple of 5, try the up and down arrows, etc.):

Quantity (between 0 and 500, should be a multiple of 5 otherwise it's invalid): 25

Source code:

```
<!DOCTYPE html>
<html>
 <head>
 <style>
   #number:invalid {
      background-color:pink;
   }
   #number:valid {
      background-color:lightGreen;
   }
 </style>
 </head>
<body>
Example of <code><input type=number></code>:<p>
   <label for="number">Quantity (between 0 and 500, should be a
multiple of 5 otherwise it's invalid): </label>
<input type="number" id="number"value="25" min="0" step="5" max="500"/>
<p>
Change the different values for attributes step, max, min, value.
Don't forget to try step="any" for float values...
</body>
</html>
```

SOURCE CODE FOR THE KNOWLEDGE CHECK

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="utf-8">
 <title>Example type=number</title>
 <style>
     #myField:valid {
         background-color:lightGreen;
     }
     #myField:invalid {
         background-color:pink;
     }
```

```
 </style>
</head>
<body>
   <label for="myField">Please enter a number between 0 and
30: </label>
   <input type="number" id="myField"min="0" step="5" max="30"/>
</body>
</html>
```

KNOWLEDGE CHECK 5.4.5 (NOT GRADED)

Suppose we enter the value 17 in the input field defined by the above code. What will the background color be?

- pink
- lightGreen
- red