{ CSS Basics }

{ CSS Basics }

CSS stands for Cascading Style Sheets. It is a language that describes the style of an HTML document.

- CSS describes how HTML elements are displayed on desktop, smartphones, tablets, paper, or in other media.
- CSS can control the layout of multiple webpages all at once.
- „Cascading“ means: More than one rule can apply to a HTML element. The rule used is chosen by cascading down from the more general rules to the specific rule.

{ CSS Basics }

CSS Syntax

```
Selector   Declaration        Declaration

   p  {  color: blue;     font-size: 11px;   }

         Property Value   Property     Value
```

- The **Selector** points to the HTML element you want to style.
- The **Declaration** block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS **Property** name and a **Value**, separated by a colon.

{ CSS Basics }

3 Ways to Insert CSS

Inline Style

```
<body style=“background-color: red;“>
```

Internal Style Sheet (inside the <head> of the HTML-page)

```
<style>
    body {background-color: red;}
</style>
```

External Style Sheet (inside the <head> of the HTML-page)

```
<head>
    <link rel=“stylesheet“ href=“my-styles.css“ type=“text/css“>
</head>
```

{ CSS Basics }

CSS Selectors

Element Selector

Select any HTML Element by name like p, h1, body, img, nav (without the <> brackets)

```
body {
    background-color: #cccccc;  /* This is a CSS comment */
    margin: 0;
    font-family: Arial, Helvetica, sans-serif;  /* inherited */
    font-size: 12px;  /* partly inherited */
    font-style: italic;  /* inherited */
}
```

In this example all the style declarations are applied to the body. Some are inherited to other HTML elements.

{ CSS Basics }

CSS Selectors

Element Selector examples ► *Child Selector*

p { … }	affects every paragraph
div p { … }	affects every paragraph inside a div element
ul li { … }	affects every list item inside an unorderd list
li { … }	affects every list item inside ul and ol
p strong { … }	affects text marked strong inside a paragraph
strong { … }	affects every text marked strong
nav ul li a { … }	affects every link inside a list item inside an unordered list inside a nav element.

{ CSS Basics }

CSS Selectors

Class Selector

Give one or more HTML Elements a class like <p class=“error-msg“...> and select this Element(s) in CSS with a **DOT** followed by the classname.

```
.error-msg {
    color: red;
    font-size: 70%;
    padding: 10px 20px;
}
```

The style declarations are applied to all HTML elements with the class „error-msg“. Use Classes for recurrent elements which should always look the same.

{ CSS Basics }

CSS Selectors

ID Selector

Give a single HTML Element an ID like <img id=“send-btn“...> and select it with # followed by this ID.

```
#send-btn {
    border: 2px solid #cccccc;
    float: left;
    width: 120px;
    margin: 10px;
}
```

The style declarations are applied only to the element with the ID „send-btn“. ID‘s should always be unique!

ID‘s can be helpful when working with JavaScript.

{ CSS Basics }

What means cascading?

Element Selector

1. p { color: black; }

Class Selector

2. p.error { color: red; }

ID Selector

3. #first { font-weight: bold; }

1. Every paragraph has a black text-color.
2. The paragraphs with the class **error** have a red text-color.
3. The paragraph with the class **error** <u>and</u> the ID **first** will be bold and red.

{ CSS Basics }

Grouping Selectors

Group Selectors to save time and to minimize the code of your stylesheet.

```
h1, h2, h3, ul, ol, .blue  {
    color: blue;
    font-family: Helvetica;
    font-weight: normal;
}
```

Separate the grouped selectors with a comma.

You can still add more styles to the elements:

```
h1 { font-size: 24px; font-weight: bold; }
h2 { font-size: 18px; }
```

{ CSS Basics }

Colors

Colors (including background colors) can be defined by:

a valid **color name**

```
    color: red;
```

a **HEX** value

```
    color: #ff0000;
```

an **RGB** value

```
    color: rgb(255 ,0, 0);
```

an **RGBa** value (transparency)

```
    background-color: rgba(255 ,0, 0, 0.5);
```

Use a color picker or color palette to find nice color combinations (i.e. https://color.adobe.com)

{ CSS Basics }

Backgrounds

CSS background properties

background-color: #ffa500;

background-image: url(bg-image.jpg);

background-position: left top; (right, bottom, center)

background-size: cover; (contain)

By default, the background-image is repeated both horizontally and vertically. To avoid this, use:

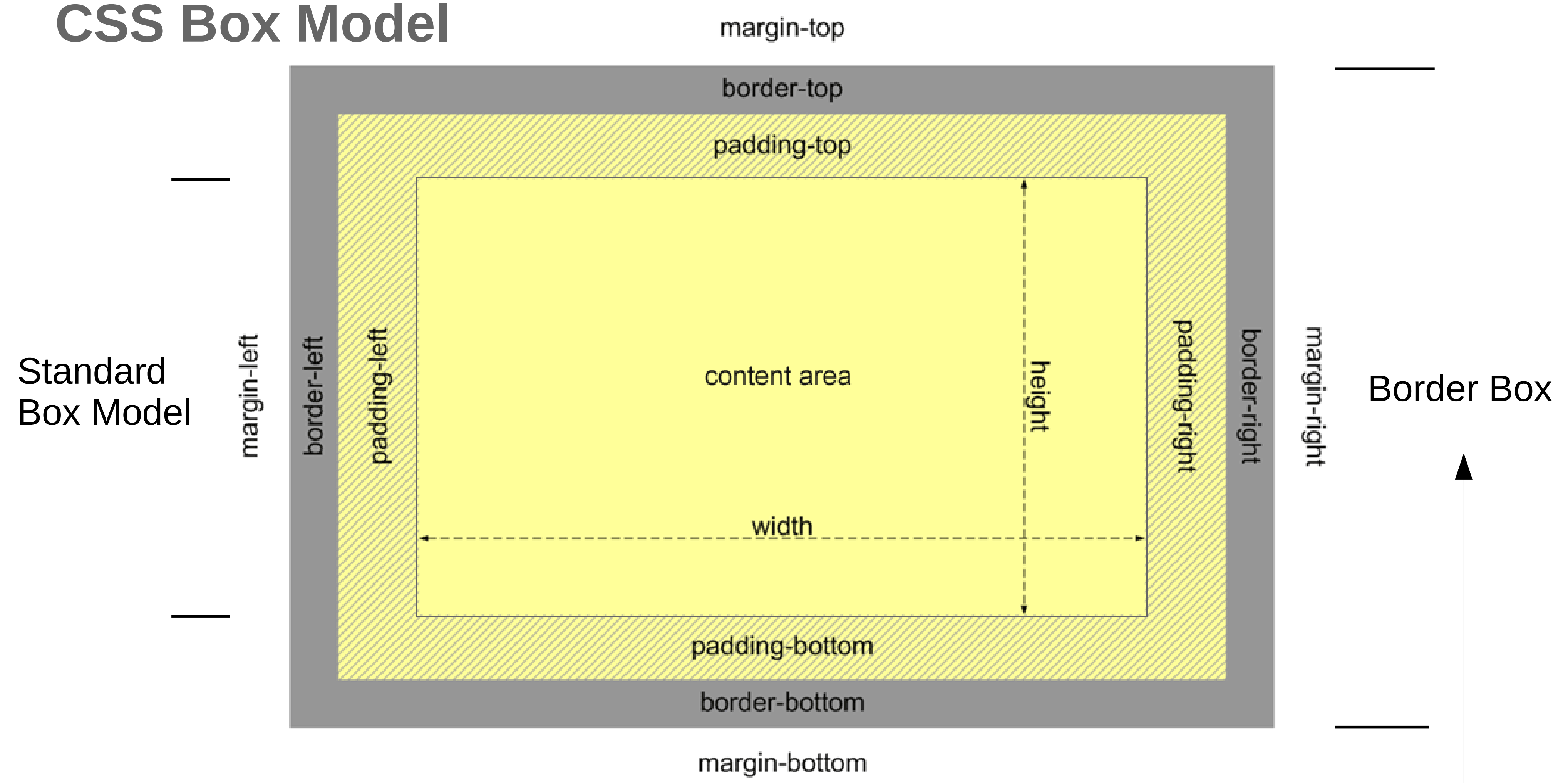
background-repeat: repeat-x; (repeat-y, no-repeat)

To fix the background-image, so it will not scroll:

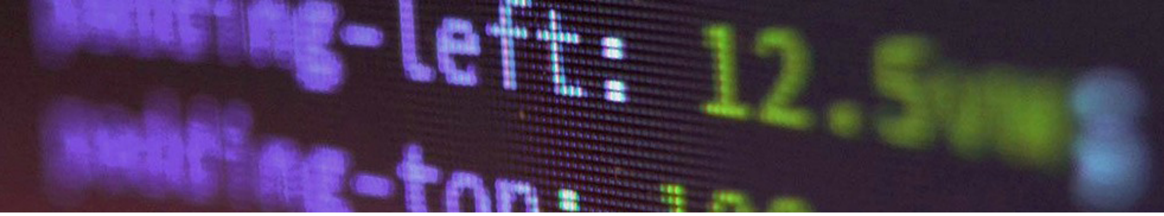
background-attachment: fixed;

{ CSS Basics }

CSS Box Model



The width and height property of an element refers to the **content area**. Alternatively use: box-sizing: border-box;



{ CSS Basics }

Borders

CSS border properties

border-style: solid; (dotted, dashed, double ...)

border-width: 2px; (in px, pt, cm, em or thin, medium, thick)

border-color: green; (valid color names, HEX, RGB)

Individual Style

border-style: dashed solid dashed solid; (top,right,bottom,left)

border-left: 5px solid red;

Shorthand properties

border: 2px dashed #333333;

{ CSS Basics }

Margins

CSS margin properties

margin-top: 40px;

margin-right: 20px;

margin-bottom: 10px;

margin-left: 20px;

Shorthand properties

margin: 40px 20px 10px 20px; (top, right, bottom, left)

margin: 20px auto; (horizontally centered)

margin: inherit; (margin inherited from the parent element)

Paddings

CSS padding properties

padding-top: 40px;

padding-right: 20px;

padding-bottom: 10px;

padding-left: 20px;

Shorthand properties

padding: 40px 20px 10px 20px; (top, right, bottom, left)

padding: 20px 30px; (top & bottom 20px, left & right 30px)

padding: inherit; (padding inherited from the parent element)

{ CSS Basics }

Width and Height

width and height properties are used to set the width and height of an element in length values like px, mm, cm or in %.

```
article {
    width: 50%;
    height: 200px;
}
```

max-width / max-height / min-width / min-height

```
article {
    max-width: 500px;
    min-height: 200px;
}
```

Remember the Box Model, when calculating the size of a box!

{ CSS Basics }

Text Formatting

color	Sets the color of text
letter-spacing	Space between characters in a text
word-spacing	Space between words in a text
line-height	Sets the line height
text-align	Horizontal alignment of text (left, right, center)
vertical-align	Vertical alignment of an element
text-transform	Controls the capitalization of text (uppercase)
text-indent	Indentation of the first line in a text-block
text-decoration	Specifies the decoration added to text

{ CSS Basics }

Font Properties

CSS font properties define the font family, boldness, size, and the style of a text.

```
p { font-family: “Times New Roman“, Arial, Verdana, sans-serif;
    font-weight: bold; (normal)
    font-size: 12px; (pt, em, %)
    font-style: italic; (normal, oblique)    }
```

If a font name contains white-space, it must be quoted.

Shorthand property

```
    font: italic bold 12px/20px Times, serif;
```

The font shorthand property defines the font-style, font-variant, font-weight, font-size/line-height, and the font-family.

{ CSS Basics }

Font Size

Font size values can be defined absolute (px, pt, cm) or relative (%, em, rem).

```
body  { font-size: 100%; }       /*   default = 16px  */
h1    { font-size: 2em; }        /*   = 32px          */
h2    { font-size: 1.5em; }      /*   = 24px          */
p     { font-size: 0.75em; }     /*   = 12px          */
```

Web Fonts

As an alternative to the default browser fonts you can use **web fonts:** https://fonts.google.com or **google-webfonts-helper** for a hassle-free way to self-host Google Fonts.

Web Fonts can increase page loading times significantly.

{ CSS Basics }

Pseudo Classes: Styling Links

Links can be styled in various ways to attract attention. Links have 4 states which can be styled different:

a:link	a normal, unvisited link
a:visited	a link the user has visited
a:hover	a link when the user mouses over it
a:active	a link the moment it is clicked

```
a:link, a:visited {
    background-color: red;
    padding: 10px 20px;
    text-align: center;
    text-decoration: none;
}
```

The :hover Pseudo Class

The Pseudo Class **:hover** can be used to change the properties of various elements, not only links.

```
img {
    width: 300px;
    border: 2px solid #666;
    margin: 20px 10px;
}

img:hover {
    width: 340px;
    border: 2px solid #333;
    box-shadow: 5px 5px 10px #222;
}
```

{ CSS Basics }

Styling Lists

Set different list item markers for ordered or unordered lists or use an image as a list item marker.

list-style	Sets all the properties for a list
list-style-type	Specifies the type of list-item marker
list-style-position	list-item markers inside or outside
list-style-image	Specifies an image as the list-item marker

```
ul {
    list-style-type: square;
    list-style-position: inside;
}
```

Shorthand property

```
ul { list-style: square inside url(“square-blue.gif“); }
```

{ CSS Basics }

Styling Lists for the Navigation

Style your list or list items with background-colors, margin, padding … to make a nice navigation.

```
nav ul { list-style-type: none; padding: 0; }

nav ul li { display: inline-block; }

nav li a {
    text-decoration: none;
    color: white;
    background-color: blue;
    padding: 8px 12px;

}

nav li a:hover { background-color: grey; }
```

{ CSS Basics }

Styling Tables

Style your <table> or <th>, <td>, <tr> elements with table properties and other styles like background, color etc.

border	All border properties in one declaration
border-collapse	borders should be collapsed or not
border-spacing	Distance between the borders of cells
caption-side	Placement of a table caption
empty-cells	Specifies whether or not to display borders and background on empty cells
table-layout	Sets the layout algorithm (auto, fixed)
tr:nth-child(even)	nth-child-selector

{ CSS Basics }

CSS Position

The position property specifies the type of positioning method used for an element (static, relative, fixed or absolute).

static	Default position
relative	Relative positioned to its normal position
absolute	Relative positioned to its parent element
fixed	Relative positioned to the viewport
sticky	Mix of position relative and fixed

```
article {
    position: relative;
    top: 20px;
    left: 50px;
}
```

{ CSS Basics }

Z-Index and Position

The z-index specifies the stack order of positioned elements.

Elements with higher z-index are on top of elements with lower z-index.

```
img.logo {
    position: absolute;
    top: 20px;
    right: 50px;
    z-index: 10;
}
```

```
button.contact {
    position: fixed;
    bottom: 20px;
    left: 50px;
    z-index: 99;
}
```

{ CSS Basics }

Float

The float property removes HTML elements from the normal flow. Its simplest use is to wrap text around images.

```
img {
    float: left;
    width: 200px;
    margin-right: 10px;
}
```

The paragraph (or other element) following the image will flow around the image on its right side with a margin of 10px.

To end the floating of elements use:

```
    clear: left; (right, both)
```

{ CSS Basics }

The display property

Every HTML element has a default display property (inline, block, table, table-cell etc.). This property can be overwritten by:

```
    display: block;   /*  inline-block for horizontal navigation  */
```

To hide an element, you can use:

```
    display: none;
```

This can be used as a starting point to show elements (like submenu) depending on a certain action (like mouse-over).

`visibility: hidden;` also hides elements, but leaves an empty space with the size of the element.

{ CSS Basics }

Horizontal & Vertical Align

To horizontally center a block element (like <div>), use
margin: 0 auto; and set a width property to the element

To horizontally center an image, use
margin: 0 auto; and display: block;

To center the text like in paragraphs or headings, use
text-align: center;

Another method for aligning elements is to use
position: absolute; inside a container with position: relative

You can use the float property for aligning left or right.

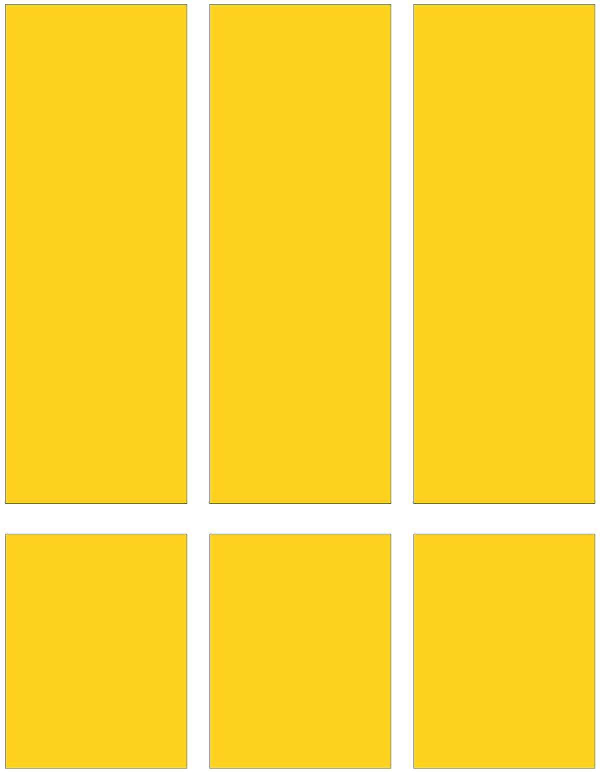
You can center elements vertically and horizontally with FLEX
justify-content: center; align-items: center;

{ CSS Basics }

Layout Concepts

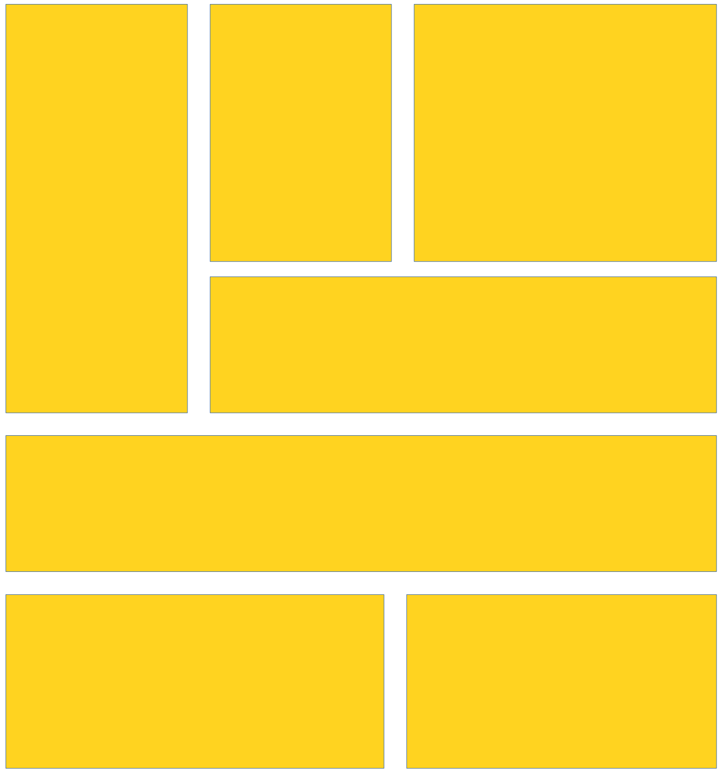
Flexbox

display: flex;



Grid

display: grid;



{ CSS Basics }

Attribute Selectors

The attribute selector is used to select elements with a specified attribute (value).

input[type="text"] { … } selects all input fields in a form with the type attribute „text“.

a[target] { … } selects all links which have a target attribute.

a[target="_blank"] { … } selects all links with a target attribute set to „_blank“.

[title="flower"] { … } selects all elements with the title attribute „flower“.

{ CSS Basics }

Styling Forms

To style form fields we can make use of the attribut selector.

```
input[type="text"] {
    width: 300px;
    height: 30px;
    margin: 10px 0;
    background: #eee;
    border: 1px solid #555;
}

input[type="submit"]  { width: 150px; }

input[type="button"]:hover  { background: #ccc; }

input[type="text"]:focus  { background: grey; }
```

CSS Combinators

A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator.

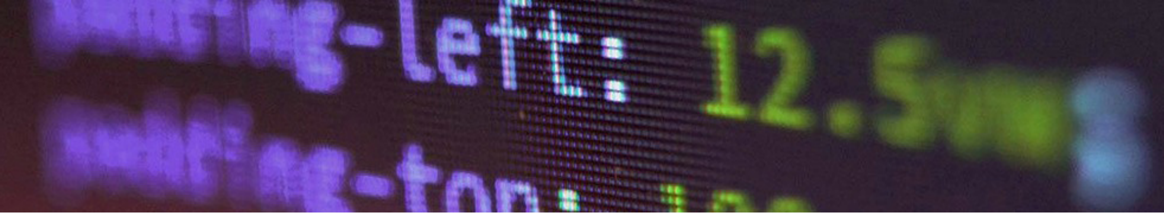
section p { … } selects all <p> elements inside <section>

section p a { … } selects all links inside a <p> element which is placed inside a section element

section > p { … } selects all <p> elements that are immediate children of the section element

section + p { … } selects all <p> elements that are placed immediately after the section element

{ CSS Basics }



CSS Media Queries

Media Queries help you to control the Layout of your website in different devices (Responsive Webdesign).

In CSS file or inside the <style> tag of your HTML document

```
@media screen and (max-width: 480px) {

    nav.mobile { display: block; }
    nav.desktop { display: none; }

}
```

Inside the link to your external Style Sheet

```
<link rel=“stylesheet“ href=“mobile-styles.css“ type=“text/css“

  media=“screen and (max-width: 480px)“ >
```

Organizing your Stylesheets

For a complex website it is often useful to have your styles organized in several stylesheets. You can import those stylesheets in your main stylesheet.

```
@import url("normalize.css");

@import url("grid-layout.css");

@import url("text-styles.css");

@import url("print.css") print;

@import url("mobile.css") screen and (max-
width:740px);

body { background-color: ...
```