

Cascading style sheets

The best way to create websites is to keep the content separate from the presentation.

The best way to create websites is to keep the content separate from the presentation. HTML is used to specify *what* goes on the page,

The best way to create websites is to keep the content separate from the presentation. HTML is used to specify *what* goes on the page, and this is separated out from *how* the page should look.

The best way to create websites is to keep the content separate from the presentation. HTML is used to specify *what* goes on the page, and this is separated out from *how* the page should look.

The presentation of the page is controlled by a *cascading style sheet* (CSS) file.

The best way to create websites is to keep the content separate from the presentation. HTML is used to specify *what* goes on the page, and this is separated out from *how* the page should look.

The presentation of the page is controlled by a *cascading style sheet* (CSS) file.

This approach makes creating pages much quicker,

The best way to create websites is to keep the content separate from the presentation. HTML is used to specify *what* goes on the page, and this is separated out from *how* the page should look.

The presentation of the page is controlled by a *cascading style sheet* (CSS) file.

This approach makes creating pages much quicker, and allows for a complete change of style easily.

If you view the source for the course webpage, you will find plain HTML

If you view the source for the course webpage, you will find plain HTML — headings, paragraphs, images and tables —

If you view the source for the course webpage, you will find plain HTML — headings, paragraphs, images and tables — but no commands for alignment, colour or size.

If you view the source for the course webpage, you will find plain HTML — headings, paragraphs, images and tables — but no commands for alignment, colour or size. Instead, there is a line in the head of the document

If you view the source for the course webpage, you will find plain HTML — headings, paragraphs, images and tables — but no commands for alignment, colour or size. Instead, there is a line in the head of the document

```
<link rel="stylesheet" type="text/css"  
href="css/course_pages.css"/>
```

If you view the source for the course webpage, you will find plain HTML — headings, paragraphs, images and tables — but no commands for alignment, colour or size. Instead, there is a line in the head of the document

```
<link rel="stylesheet" type="text/css"  
href="css/course_pages.css"/>
```

which tells the browser to use the CSS file at the relative URL `css/course_pages.css`.

The CSS file consists of a list of style specifications.

The CSS file consists of a list of style specifications.

```
html
```

```
{  
    background-color:#fff;  
}
```

```
body
```

```
{  
    font-family:"arial","helvetica";  
    color:#423;  
    background-color:#ffffe4;  
    border:#423 double;  
    width:700px;  
    ...
```

Each of these specifications takes the form

Each of these specifications takes the form

```
selector  
{  
    property:value;  
}
```

Each of these specifications takes the form

```
selector
{
    property:value;
}
```

The simplest type of selector is an *HTML selector* which corresponds to things created using the basic tags.

Each of these specifications takes the form

```
selector
{
    property:value;
}
```

The simplest type of selector is an *HTML selector* which corresponds to things created using the basic tags. The selectors `html` and `body` come in this category,

Each of these specifications takes the form

```
selector
{
    property:value;
}
```

The simplest type of selector is an *HTML selector* which corresponds to things created using the basic tags. The selectors `html` and `body` come in this category, as do `p`, `img`, `h1` and so on.

It's easy to understand what a lot of the specifications do.

It's easy to understand what a lot of the specifications do. For example, the `background-color` property changes the background colour of the element.

It's easy to understand what a lot of the specifications do. For example, the `background-color` property changes the background colour of the element. Hence,

```
html
{
    background-color:#fff;
}
```

It's easy to understand what a lot of the specifications do. For example, the `background-color` property changes the background colour of the element. Hence,

```
html
{
    background-color:#fff;
}
```

tells the browser to format everything that sits between the `<html>` and `</html>` tags with background colour `#fff`,

It's easy to understand what a lot of the specifications do. For example, the `background-color` property changes the background colour of the element. Hence,

```
html
{
    background-color:#fff;
}
```

tells the browser to format everything that sits between the `<html>` and `</html>` tags with background colour `#fff`, which means 'white' (see later).

It's easy to understand what a lot of the specifications do. For example, the `background-color` property changes the background colour of the element. Hence,

```
html
{
    background-color:#fff;
}
```

tells the browser to format everything that sits between the `<html>` and `</html>` tags with background colour `#fff`, which means 'white' (see later). This gives the course webpage its white base colour.

Similarly,

```
body
{
    font-family:"arial","helvetica";
}
```

Similarly,

```
body
{
    font-family:"arial","helvetica";
}
```

tells the browser to use the font Arial for everything in the body of the document

Similarly,

```
body
{
    font-family:"arial","helvetica";
}
```

tells the browser to use the font Arial for everything in the body of the document (that is, between the <body> and </body> tags)

Similarly,

```
body
{
    font-family:"arial","helvetica";
}
```

tells the browser to use the font Arial for everything in the body of the document (that is, between the <body> and </body> tags) and Helvetica if Arial isn't installed on the user's computer.

Similarly,

```
body
{
    font-family:"arial","helvetica";
}
```

tells the browser to use the font Arial for everything in the body of the document (that is, between the <body> and </body> tags) and Helvetica if Arial isn't installed on the user's computer.

We will play around with CSS selectors, properties and values more in the Week 9 Lab.

Colours in CSS

The standard way to specify colours in CSS is to use *hexadecimal numbers*.

Colours in CSS

The standard way to specify colours in CSS is to use *hexadecimal numbers*. Hexadecimal is the number system in base 16.

Colours in CSS

The standard way to specify colours in CSS is to use *hexadecimal numbers*. Hexadecimal is the number system in base 16. Its 16 digits are $0, \dots, 9, a, \dots, f$.

Colours in CSS

The standard way to specify colours in CSS is to use *hexadecimal numbers*. Hexadecimal is the number system in base 16. Its 16 digits are $0, \dots, 9, a, \dots, f$.

For example, the hexadecimal number $1a$ corresponds to $16 + 10 = 26$ in decimal,

Colours in CSS

The standard way to specify colours in CSS is to use *hexadecimal numbers*. Hexadecimal is the number system in base 16. Its 16 digits are $0, \dots, 9, a, \dots, f$.

For example, the hexadecimal number $1a$ corresponds to $16 + 10 = 26$ in decimal, and ff corresponds to $15 \times 16 + 15 = 255$.

Every colour can be made by mixing the three primary colours (of light) red, green and blue,

Every colour can be made by mixing the three primary colours (of light) red, green and blue, so we specify colours by giving their red, green and blue components.

Every colour can be made by mixing the three primary colours (of light) red, green and blue, so we specify colours by giving their red, green and blue components. These components can range from 0 to 255 (in decimal); that is, from 0 to *ff* (in hexadecimal).

To specify bright red, for example, we set the red component to maximum (ff)

To specify bright red, for example, we set the red component to maximum (ff) and the green and blue components to minimum (0 and 0).

To specify bright red, for example, we set the red component to maximum (*ff*) and the green and blue components to minimum (0 and 0). We encode this as the six-digit hexadecimal number

$$\underbrace{ff}_R \underbrace{00}_G \underbrace{00}_B .$$

Similarly, bright green is `00ff00`

Similarly, bright green is `00ff00` and bright blue is `0000ff`.

Similarly, bright green is `00ff00` and bright blue is `0000ff`.
White is

Similarly, bright green is $00ff00$ and bright blue is $0000ff$.
White is $ffffff$

Similarly, bright green is $00ff00$ and bright blue is $0000ff$.
White is $ffffff$ and black is

Similarly, bright green is $00ff00$ and bright blue is $0000ff$.
White is $ffffff$ and black is 000000 .

Similarly, bright green is $00ff00$ and bright blue is $0000ff$.
White is $ffffff$ and black is 000000 . The number $aaaaaa$
will be

Similarly, bright green is $00ff00$ and bright blue is $0000ff$.
White is $ffffff$ and black is 000000 . The number $aaaaaa$
will be a grey,

Similarly, bright green is $00ff00$ and bright blue is $0000ff$.
White is $ffffff$ and black is 000000 . The number $aaaaaa$
will be a grey, and 660066 will be

Similarly, bright green is $00ff00$ and bright blue is $0000ff$.
White is $ffffff$ and black is 000000 . The number $aaaaaa$
will be a grey, and 660066 will be a dark purple.

Similarly, bright green is `00ff00` and bright blue is `0000ff`.
White is `ffffff` and black is `000000`. The number `aaaaaa`
will be a grey, and `660066` will be a dark purple.

Thankfully, there are lots of colour charts in existence.

Similarly, bright green is `00ff00` and bright blue is `0000ff`. White is `ffffff` and black is `000000`. The number `aaaaaa` will be a grey, and `660066` will be a dark purple.

Thankfully, there are lots of colour charts in existence. Google 'CSS colour selector' and you will find sites that allow you to pick a colour and will return the hexadecimal code.

You may have noticed that the example we gave earlier (where *fff* represented white) doesn't quite fit this description.

You may have noticed that the example we gave earlier (where *fff* represented white) doesn't quite fit this description. In fact, *fff* is short-hand for *ffffff*.

You may have noticed that the example we gave earlier (where *fff* represented white) doesn't quite fit this description. In fact, *fff* is short-hand for *ffffff*. Similarly, *423* means *442233* and *1ae* represents *11aaee*.

You may have noticed that the example we gave earlier (where *fff* represented white) doesn't quite fit this description. In fact, *fff* is short-hand for *ffffff*. Similarly, *423* means *442233* and *1ae* represents *11aaee*.

It is fine to use either the three-digit or six-digit representations for colours as you choose.

Browsers and doctype declarations

One thing that can cause issues with CSS is that different browsers display the results slightly differently.

One thing that can cause issues with CSS is that different browsers display the results slightly differently. This is something that you can't control totally.

One thing that can cause issues with CSS is that different browsers display the results slightly differently. This is something that you can't control totally.

One thing that can make results more consistent is to declare the *doctype* at the top of the HTML file.

The doctype declaration is a line that you copy and paste into each HTML file.

The doctype declaration is a line that you copy and paste into each HTML file. It tells the browser which version of HTML you are using,

The doctype declaration is a line that you copy and paste into each HTML file. It tells the browser which version of HTML you are using, and hence which conventions to follow.

The doctype declaration is a line that you copy and paste into each HTML file. It tells the browser which version of HTML you are using, and hence which conventions to follow.

The version of HTML we have used is HTML5, and the relevant line is simply

The doctype declaration is a line that you copy and paste into each HTML file. It tells the browser which version of HTML you are using, and hence which conventions to follow.

The version of HTML we have used is HTML5, and the relevant line is simply

```
<!DOCTYPE HTML>
```

It is a good idea to put this at the top of all your HTML files.

The doctype declaration is a line that you copy and paste into each HTML file. It tells the browser which version of HTML you are using, and hence which conventions to follow.

The version of HTML we have used is HTML5, and the relevant line is simply

```
<!DOCTYPE HTML>
```

It is a good idea to put this at the top of all your HTML files. If you need to find the line to copy and paste, just view the source from the course webpage.

Webpace and domain names

So far, the webpages we have created have been *offline*,

So far, the webpages we have created have been *offline*, that is stored on just one computer.

So far, the webpages we have created have been *offline*, that is stored on just one computer. To turn the pages into part of the world-wide web requires the use of a *web server*.

A web server is a computer which allows users to connect to it and access its files.

A web server is a computer which allows users to connect to it and access its files. It is possible to set up your own computer as a web server,

A web server is a computer which allows users to connect to it and access its files. It is possible to set up your own computer as a web server, but it's more common to obtain *webspace* (that is, filespace) on a dedicated web server.

A web server is a computer which allows users to connect to it and access its files. It is possible to set up your own computer as a web server, but it's more common to obtain *webspace* (that is, filespace) on a dedicated web server.

The web server is often called the *host* of the site.

The web server (or host) that is used for the MAS115 website belongs to the university.

The web server (or host) that is used for the MAS115 website belongs to the university. The URL <http://mas115.group.shef.ac.uk> points to our filespace on the university's server.

The web server (or host) that is used for the MAS115 website belongs to the university. The URL `http://mas115.group.shef.ac.uk` points to our filespace on the university's server.

Accessing `http://mas115.group.shef.ac.uk` via a browser makes the web server return the `index.html` file inside that filespace,

The web server (or host) that is used for the MAS115 website belongs to the university. The URL `http://mas115.group.shef.ac.uk` points to our filespace on the university's server.

Accessing `http://mas115.group.shef.ac.uk` via a browser makes the web server return the `index.html` file inside that filespace, which the browser displays.

Unfortunately, the university doesn't offer webspace to students.

Unfortunately, the university doesn't offer webspace to students. However, there are many companies that offer webspace.

Unfortunately, the university doesn't offer webspace to students. However, there are many companies that offer webspace. Some offer it free in exchange for adverts on your site,

Unfortunately, the university doesn't offer webspace to students. However, there are many companies that offer webspace. Some offer it free in exchange for adverts on your site, others charge an annual fee.

Unfortunately, the university doesn't offer webspace to students. However, there are many companies that offer webspace. Some offer it free in exchange for adverts on your site, others charge an annual fee. You shouldn't have to pay much money for webspace.

File transfer protocol (FTP)

Once you have built a website and obtained webspace, you need to transfer your site to the web server.

Once you have built a website and obtained webspace, you need to transfer your site to the web server. The most common method of doing this is using *file transfer protocol* (FTP).

Once you have built a website and obtained webspace, you need to transfer your site to the web server. The most common method of doing this is using *file transfer protocol* (FTP).

Using FTP is easy.

Once you have built a website and obtained webspace, you need to transfer your site to the web server. The most common method of doing this is using *file transfer protocol* (FTP).

Using FTP is easy. The most usual approach is to use an *FTP client* (for example, Filezilla).

Once you have built a website and obtained webspace, you need to transfer your site to the web server. The most common method of doing this is using *file transfer protocol* (FTP).

Using FTP is easy. The most usual approach is to use an *FTP client* (for example, Filezilla). After typing in the log-in details for your webspace, transferring the files is a simple case of drag-and-drop.

Once you have built a website and obtained webspace, you need to transfer your site to the web server. The most common method of doing this is using *file transfer protocol* (FTP).

Using FTP is easy. The most usual approach is to use an *FTP client* (for example, Filezilla). After typing in the log-in details for your webspace, transferring the files is a simple case of drag-and-drop.

As soon as the files are onto the web server they will be accessible via the world-wide web.

FileZilla window: sam-marsh.staff.shef.ac.uk - sm1sm@staff.shef.ac.uk - FileZilla

File Edit View Transfer Server Bookmarks Help

Status: Directory listing successful
Command: DELE Thumbs.db
Response: 250 Deleted Thumbs.db

Local site: broot\websites\mas115\ Remote site: /www/mas115

Local site tree:
mas114
mas115
 docs
 uploads
 shef-only

Remote site tree:
mas114
mas115
 docs
 uploads
 shef-only

Local site file list:
..
docs
uploads
favicon.ico
index.php
mas115.jpg

Remote site file list:
..
docs
uploads
favicon.ico
index.php
mas115.jpg

Local site summary: 3 files and 2 directories. Total size: 45,532

Remote site summary: 3 files and 2 directories. Total size: 45,532 bytes

Domain names

The final step to running a website is to obtain a *domain name* (such as `bbc.co.uk` or `channel14.com`).

The final step to running a website is to obtain a *domain name* (such as `bbc.co.uk` or `channel14.com`). There are lots of companies that will register domain names online and the registration process is relatively simple.

The final step to running a website is to obtain a *domain name* (such as `bbc.co.uk` or `channel14.com`). There are lots of companies that will register domain names online and the registration process is relatively simple.

After registering your domain name, it needs to be configured to point to your webspace.

The final step to running a website is to obtain a *domain name* (such as `bbc.co.uk` or `channel14.com`). There are lots of companies that will register domain names online and the registration process is relatively simple.

After registering your domain name, it needs to be configured to point to your webspace. Once this is done, your website will be complete!

The final step to running a website is to obtain a *domain name* (such as `bbc.co.uk` or `channel14.com`). There are lots of companies that will register domain names online and the registration process is relatively simple.

After registering your domain name, it needs to be configured to point to your webspace. Once this is done, your website will be complete!

Domain names are not free, but most of them are cheap.

The final step to running a website is to obtain a *domain name* (such as `bbc.co.uk` or `channel14.com`). There are lots of companies that will register domain names online and the registration process is relatively simple.

After registering your domain name, it needs to be configured to point to your webspace. Once this is done, your website will be complete!

Domain names are not free, but most of them are cheap. There is an annual fee for renewing the domain name, but again this is small.

Important! Two more videos to watch

- Instructions for the mini-project peer assessment
- Demonstration of software options for working on HTML

About Computer Lab 8

In Computer Lab 8 we will experiment further with CSS, getting used to affecting the layout of pages.

Mini-project peer assessment

Now that all projects have been submitted, it is your job to read and mark some of your peers' projects as part of the peer assessment process. Each student has been assigned 3-4 projects to assess. To view the projects you will assess, log into the upload system:

<http://somas-uploads.shef.ac.uk/mas115>

MAS115: Project Uploads

Note: to download Python or R files, right-click, then save-as.

Mini Project 1	Thu 19 Nov, 01:00 am	Upload files	View pdf tex py			
----------------	----------------------	------------------------------	---------------------------------	--	--	--

TO MARK			
	View file pdf tex py	Leave feedback	
	View file pdf tex py	Leave feedback	
	View file pdf tex py	Leave feedback	
	View file pdf tex py	Leave feedback	

[MAS115 module webpage](#)

MAS115: Peer feedback

You should leave feedback on Mini Project 1 for below. You should comment on each of the programming, the extension, the typesetting and the standard of writing. These comments will be read by the student.

What do you think was good about this project?

What do you think could be improved?

What mark (out of 4) would you give this project?

[Save peer feedback](#)

[Save and view peer feedback](#)

[MAS115 module webpage](#)

Here's what you need to do:

- Read through the PDFs.

Here's what you need to do:

- Read through the PDFs.
- Run the Python script by downloading it (right-click, save-as), opening in a text editor (e.g. Notepad), and pasting the text into a new Google Colab notebook. Alternatively, you can use Spyder if you've installed it, or else an online Python interpreter (e.g. repl.it).

Here's what you need to do:

- Read through the PDFs.
- Run the Python script by downloading it (right-click, save-as), opening in a text editor (e.g. Notepad), and pasting the text into a new Google Colab notebook. Alternatively, you can use Spyder if you've installed it, or else an online Python interpreter (e.g. repl.it).
- You can look at the LaTeX file if you spot a problem, or are interested.

Here's what you need to do:

- Read through the PDFs.
- Run the Python script by downloading it (right-click, save-as), opening in a text editor (e.g. Notepad), and pasting the text into a new Google Colab notebook. Alternatively, you can use Spyder if you've installed it, or else an online Python interpreter (e.g. repl.it).
- You can look at the LaTeX file if you spot a problem, or are interested.
- You must leave comments on what was good about the project and what could be improved in the upload system.

Here's what you need to do:

- Read through the PDFs.
- Run the Python script by downloading it (right-click, save-as), opening in a text editor (e.g. Notepad), and pasting the text into a new Google Colab notebook. Alternatively, you can use Spyder if you've installed it, or else an online Python interpreter (e.g. repl.it).
- You can look at the LaTeX file if you spot a problem, or are interested.
- You must leave comments on what was good about the project and what could be improved in the upload system.
- Give a mark out of 4 to each project.

- You will receive 2 marks for taking part in the peer assessment. To receive these marks, you must complete the comments and provide a mark for each project online. Your comments must take the form of one or more full sentences and demonstrate that you have properly engaged with the process. You must not leave the same comment for more than one project.

- You will receive 2 marks for taking part in the peer assessment. To receive these marks, you must complete the comments and provide a mark for each project online. Your comments must take the form of one or more full sentences and demonstrate that you have properly engaged with the process. You must not leave the same comment for more than one project.
- The deadline for submitting the comments is midnight at the end of Friday 4 December (Week 10).

To help you to decide on marks, you can find marking descriptors on page 3 of the project description. The following points may also help you decide:

- Which project(s) did you like the best?

To help you to decide on marks, you can find marking descriptors on page 3 of the project description. The following points may also help you decide:

- Which project(s) did you like the best?
- How well written and explained was the code? Did it work?

To help you to decide on marks, you can find marking descriptors on page 3 of the project description. The following points may also help you decide:

- Which project(s) did you like the best?
- How well written and explained was the code? Did it work?
- How well was the project extended?

To help you to decide on marks, you can find marking descriptors on page 3 of the project description. The following points may also help you decide:

- Which project(s) did you like the best?
- How well written and explained was the code? Did it work?
- How well was the project extended?
- Did the author use LaTeX well?

To help you to decide on marks, you can find marking descriptors on page 3 of the project description. The following points may also help you decide:

- Which project(s) did you like the best?
- How well written and explained was the code? Did it work?
- How well was the project extended?
- Did the author use LaTeX well?
- Was the writing fluent and free from spelling mistakes?

- Perhaps use PMI: plus points, minus points, interesting points.

- Perhaps use PMI: plus points, minus points, interesting points.
- Broadly speaking, 0 corresponds to a non-submission, and 4 corresponds to a flawless piece of work that comprehensively addresses the project brief.

- Perhaps use PMI: plus points, minus points, interesting points.
- Broadly speaking, 0 corresponds to a non-submission, and 4 corresponds to a flawless piece of work that comprehensively addresses the project brief.
- Please don't be too harsh on the author. They may not speak English as a first language or may be struggling with computer programming.

- Perhaps use PMI: plus points, minus points, interesting points.
- Broadly speaking, 0 corresponds to a non-submission, and 4 corresponds to a flawless piece of work that comprehensively addresses the project brief.
- Please don't be too harsh on the author. They may not speak English as a first language or may be struggling with computer programming.
- Some students may attach a 'yellow sticker' to their work to signify a specific learning disability such as dyslexia. This means the work should be marked for content as normal, but feedback should be chosen appropriately. The upload system should make such students clear.

Please take this seriously! The peer assessment class is one of the most valuable parts of this module. As we have explained in previous lectures, it gives you an invaluable chance to look at other people's work and learn from your peers, something that you wouldn't get from us marking your work.

If you are unsure about what you are supposed to do, please post on the online discussion board.