</> Saving your work

Introduksjon

It’s important to **save** your work regularly, so you don’t lose any of th
at where to save your code and various other things you might need (
correctly.

Previewing the changes yo

When you make changes in the **text editor**, you have to **save** them
you’ve saved them, you have to **refresh** the browser. If you’re on Wir
CTRL+S to save the file you’re editing, and many browsers will allow
key (if using Internet Explorer). On the Mac these shortcuts are simila
command+R to refresh the page in the browser.

Editing pages as you go

But what if you want to try something really quickly without switching
developer tools before, then forgive us explaining how to do it again,

1. Go to your **web browser**.
2. Right click anything interesting, and then click **Inspect elemen**
   up. It will show you the page’s code as the same time as showin
3. Move your mouse over different pieces of code. The correspondi
   see which bit does what.
4. On the left, the panel shows you the **HTML structure** of your d
   will see which bits of **CSS** this element uses. This is useful when
   all the CSS you’ve written.
5. **Double-clicking** on the **HTML** or **single-clicking** on **CSS** allow
   immediately, so you can clearly see their effect.

6. The catch is that these changes aren't actually **saved** – they are
back to your document you can see that your code hasn't chang
in or copy them from the developer panel.

Saving things in the right p

When your site needs only one file – the HTML document – then your
memory stick, move it inside directories to organise it, etc. and it still
including pictures, or audio clips, videos and the like, you need to rem

Case #1 – all files live in the sam

Let's say you have your page.html on the desktop, and it includes a p
`alt="A kitty playing with some wool.">` The browser will look for t
our case the desktop (desktop is just a directory like any other, only a

Case #2 – going into different di

What if we wanted to copy our site onto a memory stick? We would ha
same directory on the stick. Maybe we can make it easier by putting t
structure now looks like this:

```
my_kitty_site
|
\- page.html
|
\ kitty.jpg
```

Let's tidy it up a bit more. What if later we want to include more imag
to find them.

```
my_kitty_site
|
\- page.html
|
\ - images |
  \ kitty.jpg
```

```
---
  \ kitty_and_puppy.jpg
```

Now we we open up the page in the browser, the image doesn't show
help it by adding what we call a path – a route through the directories
our case we would have to go into images directory before we get to t
`alt="A kitty playing with some wool.">` The images/ bit before the
into the images directory first. Save, refresh, and the kitty is back

Case #3 - coming out of director

What if you fancied tidying up your file structure some more and put a
we did with images?

Let's say you created the following:

```
my_kitty_site
|
\ - pages
---
| \ - page.html
|
\ - images
---
  \ kitty.jpg
---
  \ kitty_and_puppy.jpg
```

Again, when we open up the page.html our images don't work. This is
page you're looking at. When we said: `<img src="images/kitty.jpg"`
meant: "from page.html go into directory called images where you wi
directory, where page.html lives! What we wanted to say was: "from p
called images where you will find kitty.jpg". So how can we say it? `<i`
`with some wool.">` ../ means "go one directory up". The value of the
these parts: ../ – go one directory up images/ – go into the images dire

Things to try

A). Organise your files however you like. Try to figure out how to make
wanted the browser to look for a file two directories up? How would yo