Dynamic Web Design

Basic concepts

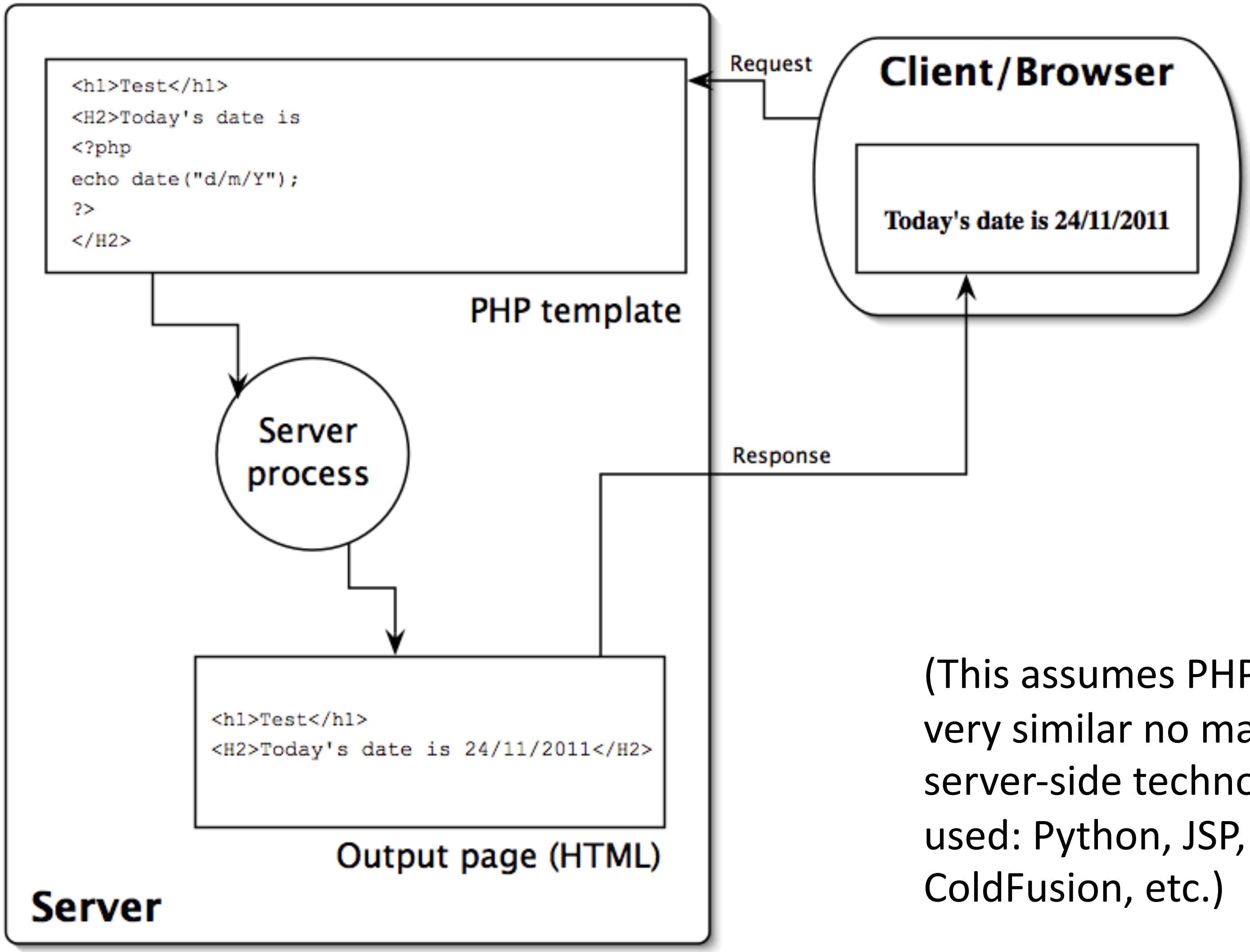
Dynamic Web Design

- What is it?
- The design of web pages that are not *static*
- I.e. pages where the *content* changes depending on something outside the page
- I say here "content", not "appearance" …
  - … because appearance can change even if the content doesn't …
  - e.g. if the content is *active* javascript/CSS

The traditional picture

- A server serves pages, usually HTML content (with CSS etc.)
  - the HTML comes from a static file, or
  - the server creates the HTML *dynamically*, by running some kind of program or script ***at the server-side***
  - (this might involve e.g. embedding data from a database, as in catalogues etc.)
- Active content might exist for rollovers, popups etc., but works ***at the client-side*** (or *browser-side*) as it would on a static page …
- To update the content, you *reload the page*
  - which re-runs the program/script
  - and might include different data into the page
- Server can receive *user data* included in the *request* for the page
  - e.g. on the URL; or if a form has been completed; or from cookies etc.
  - data is then passed to the program/script; might change the output

The basic process



```
<h1>Test</h1>
<H2>Today's date is
<?php
echo date("d/m/Y");
?>
</H2>
```

```
<h1>Test</h1>
<H2>Today's date is 24/11/2011</H2>
```

Today's date is 24/11/2011

(This assumes PHP, but it’s very similar no matter what server-side technology is used: Python, JSP, ASP, ColdFusion, etc.)

AJAX

- Nowadays (since early 2000s) the picture is usually much more complicated, because of *AJAX*
  - (Asynchronous Javascript And XML)
  - active content can make requests to the server to acquire new data or new chunks of content, and change the page *without reloading it*
  - can re-write content “on the fly”, and hence do more or less anything to a web page
  - combines well with persistent data storage in cookies
  - heavily used by Google, Amazon, etc., etc.

Design considerations

- Appearance: layout, colours, style, fonts, etc.
  - the same things you looked at in Intro to Digital Design
- Behaviour: in several loose categories
  1. Navigation (how you get from page to page)
     - usually based on following HTML links
  2. Popups (showing content in supplementary windows; often blocked!)
  3. Visual feedback (rollovers, form completion error warnings, etc.)
  4. AJAX feedback (search string completion, tag lists, etc.)
  5. Response to data input
     - when forms submitted
       (usually a case of 1, even if it doesn't always look like it …)
     - through AJAX (which might be handled in many different ways)
  6. Embedded material, often interactive, using browser plugins etc.
     (e.g. Flash, Quicktime, and other rich media types)
  7. (Increasingly) Similar to 6 but using Javascript/HTML5 in the browser

Designing behaviour

- Need to consider the *“user experience”* (UX)
- What does the user *want* and *expect*?
- How will the user *feel* about what happens at the interface?
- What do *you* want the user to do? (What is the user’s *“task”*?)
- Can you accurately *predict* what the user will do?
- Can the user accurately predict what the interface will do?
- What happens in the case of the unexpected? Is the user protected?
  - long delays in response due to systems or network
  - failure of data sources, servers, etc.
  - unexpected data retrieved, …

Some examples

- Our “people” page
- Our WIP page
  - a typical *WordPress* site
  - Content Management Systems are always dynamic
- Amazon
- Google
- Google maps
- Youtube
- YouTute

Exercise (for later)

- See if you can find some web sites that are (as far as you can see):
  - Static (though possibly including browser-based interactive elements)
  - Dynamic, but not AJAX
  - AJAX
- See if you can describe the user's "task" (even if it's just to have fun!)
- Describe how the behaviour of the site is supposed to support the task
- Estimate how effectively the design creates a UX that facilitates the task
  - Does the behaviour match the task?
  - Is the site too complex? Too simple? Just right? Just *not* right?
- Consider how the behaviour relates to the other aspects of the design (appearance, style, etc.):
  - Could you probably change all the CSS to create a completely different style, without changing the behaviour? (Cf. changing a Wordpress theme.)
  - If you did, would it make any difference to the UX?
  - How do all the elements of the design work together to create the UX?
  - How do they contribute to the *quality* of the UX?
- Have a look at some sources of advice and reflection (there are many)
  - e.g http://www.uie.com/articles/