

=====

Web scripting (more in CSC 174 and 210)

cgi-bin

CGI scripts (common gateway interface)

```
<form action="path name of CGI script" method="post">
  or "get"
<input name="val" size=3>
...    <!-- arbitrary number of these -->
<input type="submit" value="button label">
</form>
```

post

- delivers name=val pairs on stdin
- can be numerous
- not visible to user

key=val, key=val, ...

get

- delivers name=val pairs as part of URL
- can be bookmarked

script output is displayed in browser

disadvantages

- start-up (loading) cost of script
- no sandboxing of script behavior; has to be trusted
- dynamic nature visible to user
- need to print all the boilerplate HTML

embedded scripts

- require server to look inside page source

- existence **invisible** to user

PHP dominates

- Perl derivative -- built-in support for
 - access to script args (\$_REQUEST).
 - interaction with numerous database systems
 - email
 - MIME encoding
 - security and authentication
 - URI manipulation

delimit with <?php ... ?>

simple CGI replacement

- don't have to "print" the HTML parts

self-posting

- action attribute names self; script looks to see whether it has a full set of args; prints original form if no, results if yes

client-side scripts

- don't pay Internet overhead; can be much more interactive; also reduce load on server (even, e.g., for checking of input parameters)

JavaScript dominates -- built into almost every browser

- mostly but not completely standard

if you want local-only processing of form

- hijack submit behavior in <form> itself
- specify function to invoke in appropriate <input> element
- specify code of function in header of HTML file
- grab arguments from and print results to fields of 'document' object, using names defined in the HTML document object model (DOM)

Applets

- often in Java; also in Flash, Windows Media Player, PDF, other plug-ins

- give a separate script complete control of some subset of window real estate, unmediated by HTML

example: <embed type="application/x-java-applet" code="Clock.class">

- require sandboxing

- deprecated several years ago

XML

- medium-complexity markup language (simpler than SGML, more complex than HTML)

- more regular and structured than HTML

- syntax without semantics (unlike HTML)

- meant for representing all sorts of structured data

- requires semantic interpretation and formatting for display

- natural tree structure

XHTML

- XML-compliant dialect of HTML

- once proposed as a successor to HTML; now sort of a subset of HTML5

- (the world seems to have given up on deprecating old pages)

- well-formed: end tags for everything (or <tag /> singletons)

XSLT

- for processing XML

- pattern-match against trees

- XPATH provides naming conventions for elements of XML trees

- XSLT transforms XML into more XML, HTML, or whatever you want

- natural control flow is depth-first traversal of tree, but

- straightforward to program alternatives

- every template, when applied, has a notion of current node;

- naming is typically relative to that node

- XSL-FO can be used to display XML