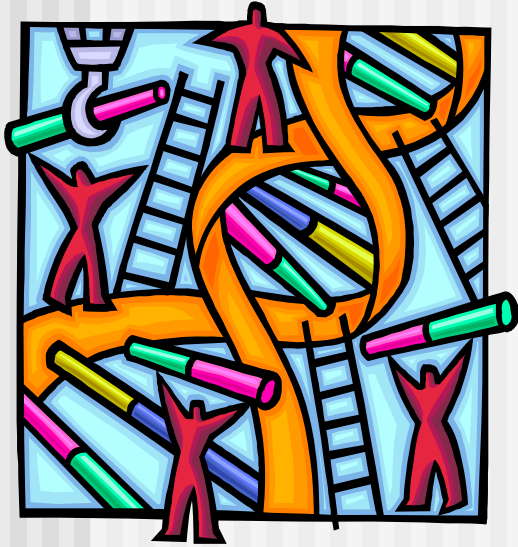


Computing Concepts for Bioinformatics

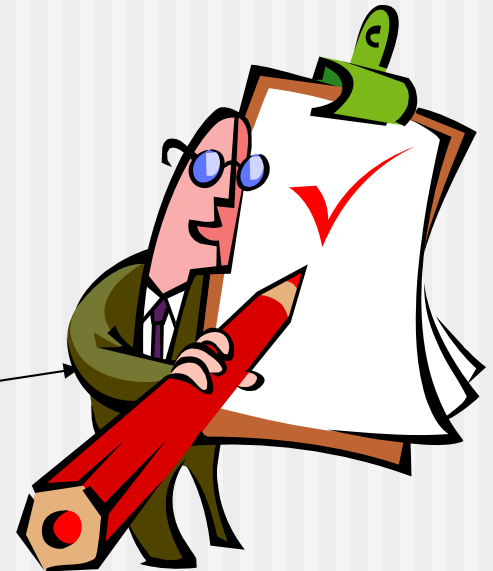
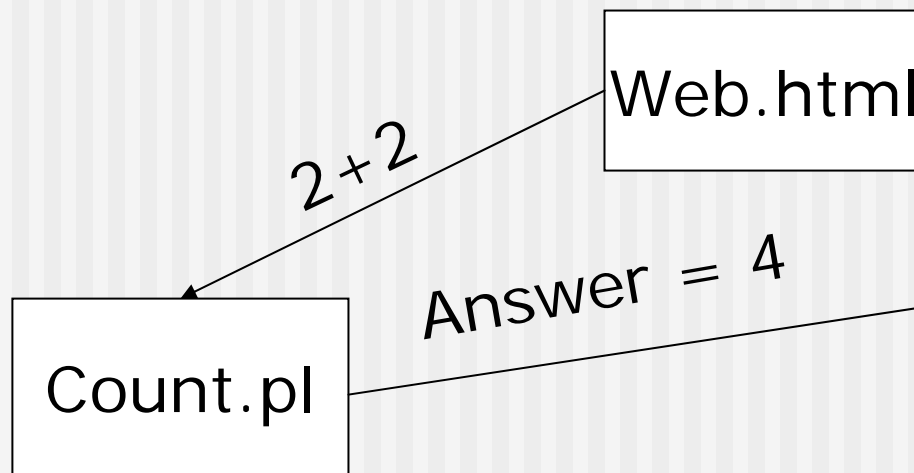


- Introduction to HTML and PHP
- Introduction to XML and web services
- Working Excel/Openoffice, Access and MySQL

<http://amadeus.biosci.arizona.edu/~nirav>

Static V/s Dynamic pages

- What are static pages
- Can you name some dynamic sites ?
- When to use static ?
- Dynamic pages are a front end to "programs/utilities"



What is HTML

- HyperText **Mark-up** Language
- It regular text with **TAGS** (also know as html tags)
- Most tags have a start and end.
`` This is bold ``
- Some tags don't have ends
`<hr>` or `
`
- The page should start with `<html>` and end with `</html>` (the file names also end in .html e.g. webgc.html)
- For further reading check:
<http://www.htmlprimer.com/>
<http://archive.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimerAll.html>

Your first web page !

- In your **home** directory
- Create a **new** directory **public_html**
- Open nedit create this page, save it as **plain.html**

```
<HTML >

<HEAD >
<TITLE > THE COOLEST PAGE ON THE WEB </TITLE >
</HEAD >

<BODY >
Welcome to my page
<b>This is boring</b> <p >
Hope you find what you are looking for !<br >
-Thanks for visiting
</BODY >

</HTML >
```

Then point your browser to:

<http://amadeus.biosci.arizona.edu/~igert##/plain.html>

XML: Why should you care ?

- eXtensible Markup Language
- W3C (World Wide Web Consortium)-endorsed standard
- Shares lineage with SGML (Standard Generalized Markup Language), which was developed in the early 1980s and widely used for large documentation projects.
- The designers of XML adopted parts of SGML and HTML to produce regular and simple-to-use language.
- It is truly extensible and we will see why !

XML: : About

- Primary use for XML is structuring data such as spreadsheets for financial and genealogical data
- Its strength lies in the fact that it is a meta-markup language, allowing one to **create new tags** to define and structure a variety of data.
- HTML has a **limited collection of predefined tags**, specifies what each tag and attribute means, and how the text between the tags should appear in the Web browser
- In contrast, XML uses tags **only to delimit** pieces of data, leaving the interpretation and rendering of data to the application that uses it.

XML: : Where do you see it ?

- Excellent for exchanging information.
- Ubiquitous in daily use from Bank transactions to inter library loans
- Many sequence analysis programs like BLAST and FASTA now produce results in XML format.
- NCBI E-utils and similar services offered by the cancer Bioinformatics Infrastructure Objects (caBIO; <http://ncicb.nci.nih.gov/core/caBIO>) , XEMBL (<http://www.ebi.ac.uk/xembl>) , DNA Data Bank of Japan (DDBJ; <http://xml.ddbj.nig.ac.jp/>) make extensive use of the XML format.

XML::Uses

- specialized mark-up formats such as MAGE-ML to describe gene expression microarray designs, manufacturing information, experimental setup, data, and data analysis results (<http://www.mged.org>).
- The Berkeley Drosophila Genome Project provides sequence annotation in GAME XML format.
- The Protein Information Resources (PIR) database can be downloaded in XML
- Gene Ontology Consortium distributes data in XML format: (<http://www.geneontology.org/GO.format.html#XML>).

XML: : Are you convinced

- A basic understanding of XML format and how to parse it is essential for using resources available from various data providers efficiently
- XML is fast becoming the de facto standard for information interchange.
- A good resource for learning more about these technologies and their application is <http://www.xml.com>
<http://www.w3schools.com/xml/>

XML::Basics

```
<?xml version="1.0"?>
<eSearchResult>
  <Count>39655</Count>
  <RetMax>10</RetMax>
  <RetStart>0</RetStart>
  <IdList Date="03/03/2004">
    <Id>15045584</Id>
    <Id>15045188</Id>
    <Id>15044961</Id>
    <Id>15044850</Id>
    <Id>15044710</Id>
    <Id>15044627</Id>
    <Id>15044396</Id>
    <Id>15044326</Id>
    <Id>15043402</Id>
    <Id>15043344</Id>
  </IdList>
</eSearchResult>
```

- First line is the XML declaration
- It defines the XML version of the document and is required.
- The next line defines the first element or the root of the document. In this example, it is **<eSearchResult>**.
- The next four lines are the child elements: **Count**, **RetMax**, **RetStart**, and **IdList**
- The Idlist element has a Date attribute and multiple Id subelements.
- The path to Count is:
/eSearchResults/Count
- The path to Id is:
/eSearchResults/IdList/
Id[2]

XML: : Basic Rules

- All tags are case sensitive.
- All elements must have a matching closing tag.
- All elements must be correctly nested.
- All attributes values must be quoted (e.g., `Date="03/03/2004"` or `version="1.0"`).
- When the above-mentioned basic rules are followed, they result in a "well-formed" XML document.

XML::DTD

- XML documents can include the DTD (Document Type Definition) or a reference to it.
- DTD is the blueprint of how data is structured within the XML document, with information about what type of values each element can contain.
- XML document can be tested for this structural integrity against the DTD specification and is considered “valid” if it meets the criteria defined in the DTD
- DTD verification is an important step when data are being exchanged between systems, thus ensuring data integrity.

XML::XSL

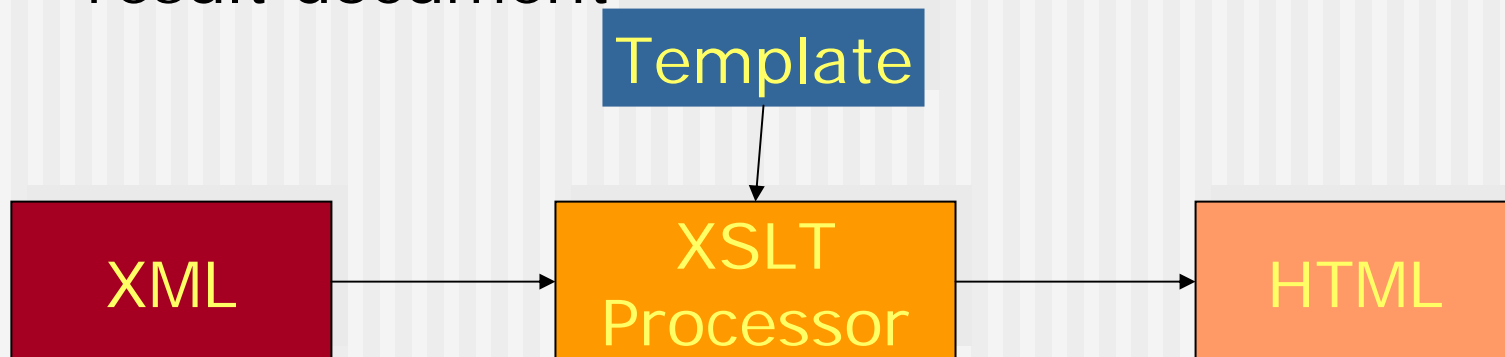
- XSL stands for eXtensible Stylesheet Language
- CSS is Cascading Style Sheets
- HTML uses predefined tags and the meanings of tags are well understood.
- The `<table>` element defines a table and a browser knows how to display it.
- CSS allow styles to HTML elements telling a browser to display an element in a special font or color
- XSL is the CSS for XML and much more

XSL::Parts

- **XSLT** is a language for transforming XML documents (XML -> HTML)
- **XPath** is a language for defining parts of an XML document (Conditional display)
- **XSL-FO** is a language for formatting XML documents (To phone, voice, images)

XSLT

- Uses a predefined template of instructions
- During transformation process, XSLT uses XPath to define parts of the XML document that match parts of the template.
- When a match is found, XSLT will transform the matching part of the source document into the result document
- The parts of the source document that do not match a template will end up unmodified in the result document



XSL: : Fun with blast xml output

- /home/student/2005/eeb/xml-files/
- Copy **blast-output.xml** and **xslt-template.xml** files from there to your public_html directory
- Open blast-output.xml in your web browser
<http://ama...../~igertxx/blast-output.xml>
- Open **xslt-template.xml** in your web browser

XSL

- Edit (using nedit) blast-output.xml and add this single line

```
<?xml version="1.0"?>  
<!DOCTYPE BlastOutput PUBLIC "-//NCBI//NCBI BlastOutput/EN" "  
<?xml-stylesheet type="text/xsl" href="xslt-template.xml" ?>  
<BlastOutput>
```

- Reload page !!
- xsltproc can use templates and convert many files at a time

XSLT :: template

```
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>My Top Blast HITs</h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th>Number      </th>
        <th>Hit</th>
        <th>Length</th>
      </tr>
      <xsl:for-each
        select="BlastOutput/BlastOutput_iterations/Iteration/Iteration_hits/Hit">
        <tr>
          <td><xsl:value-of select="Hit_num"/></td>
          <td><xsl:value-of select="Hit_id"/></td>
          <td><xsl:value-of select="Hit_len"/></td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

XML::Blast and Excel !

	A	B	D	E	F	G	H	I	J	K	L	M	N	P
1	Bla	BlastOutput_version	BlastOut	BlastOut	BlastOutput_query-d	Bla	Par	Para	Pa	Pa	Pa	Pa	Pa	Hit id
2	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc KLEBSIELLA_PI
3	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc KLEBSIELLA_PI
4	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc KLEBSIELLA_PI
5	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc SERRATIA_MAF
6	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc SERRATIA_MAF
7	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc ESCHERICHIA_C
8	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc ESCHERICHIA_C
9	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc YERSINIA_PES
10	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc YERSINIA_PES
11	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc PROTEUS_VULC
12	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc PROTEUS_VULC
13	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc PROTEUS_VULC
14	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc GH_SYMBIONT
15	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc GH_SYMBIONT
16	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc GH_SYMBIONT
17	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc HC_SYMBIONT
18	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc HC_SYMBIONT
19	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10		-3	5	2	D	1	lc HC_SYMBIONT
20	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc HC_SYMBIONT
21	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc HC_SYMBIONT
22	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc HL_SYMBIONT
23	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc HL_SYMBIONT
24	blastn	blastn 2.2.9 [May-01-2004]	MYSTUFF	lc QUERY	KLEBSIELLA_PNEUMON	1485	10	1	-3	5	2	D	1	lc HL_SYMBIONT

If you have Excel 2003 or higher
 Goto Data->XML-> XML Import and choose XML file
 (you can easily get one from pubmed by choosing
 summary type XML)

PHP: Hypertext Preprocessor

- Widely-used general-purpose scripting language
- Especially suited for Web development
- Can be embedded into HTML
- Syntax is very similar to perl
- <http://www.php.net>

Building count.php

- In your public_html using the editor create the file count.php

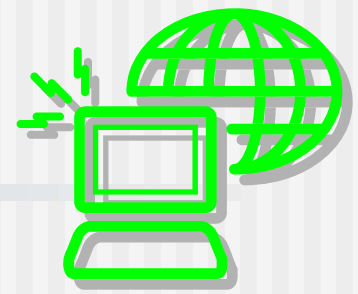
```
<?
print "You said $howmuch<br>";
print "Counting from 1 to $howmuch <br>";

for($i=1;$i<=$howmuch;$i++) {
print "$i<br>";
    }

?>
```

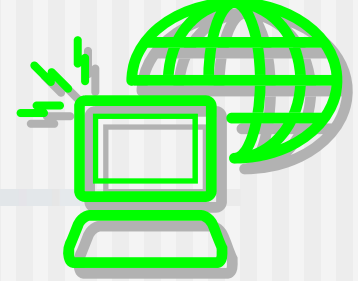
- Now point your browser to <http://amadeus.biosci.arizona.edu/~igertXX/count.php>
- Now try <http://amadeus.biosci.arizona.edu/~igertXX/count.php?howmuch=10>

Writing the web page



- The form tag allows you to define what variables will be send to which program
- `<form method = "post" action="./count.php">`
Here the action field defines which program to run.
- Within the form tags you can define fields for users to enter data I.e some values

Web page



- Create a file called `count.html` in your `public_html` directory

```
<html>
```

```
<head> This is a web interface to count down program
```

```
</head>
```

```
<body>
```

```
<form method = "post" action = "./count.php" >
```

```
<br >
```

```
<input Type=text name=howmuch >
```

```
<input Type=submit >
```

```
<input type=reset > <br >
```

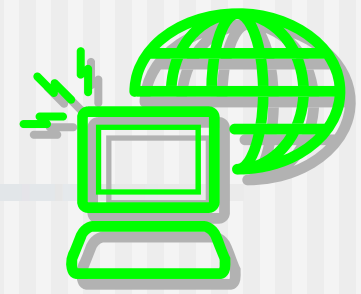
```
</form > </body > </html >
```

- Test it:

```
http://amadeus.biosci.arizona.edu/~igertXX/count.html
```



Security



- Denial of service attacks
 - Uploading large files
 - Putting web server into a loop making it busy
- Execution of unintended programs
 - Overwriting files
 - Running unwanted program like `rm`
 - Boils down to bad programming



Running a external program

- Use php system call

```
<?
//This php script runs the cal program
// with given args and displays output to web
print "<pre>";
system("/bin/cal $args");
print "</pre>";

?>
```

- <http://amadeus.biosci.arizona.edu/~login/cal.php?args=2009>
- In general avoid doing this if you can !!!

Using Excel and the web

- Create a simple php script xl.php

```
<?
for($i=1;$i<=$showmuch;$i++) {
print "$i\n";
}
?>
```

- Open Excel under data click on "get external data", "new web query"

Excel (ver. older than 2003)

New Web Query [?] [X]

1. Enter the address for the Web page that contains the data you want. If browsing, switch back to Excel once you have located the Web page in your browser.

2. Choose the part of the Web page that contains the data you want. Note that pre-formatted sections are treated as tables.

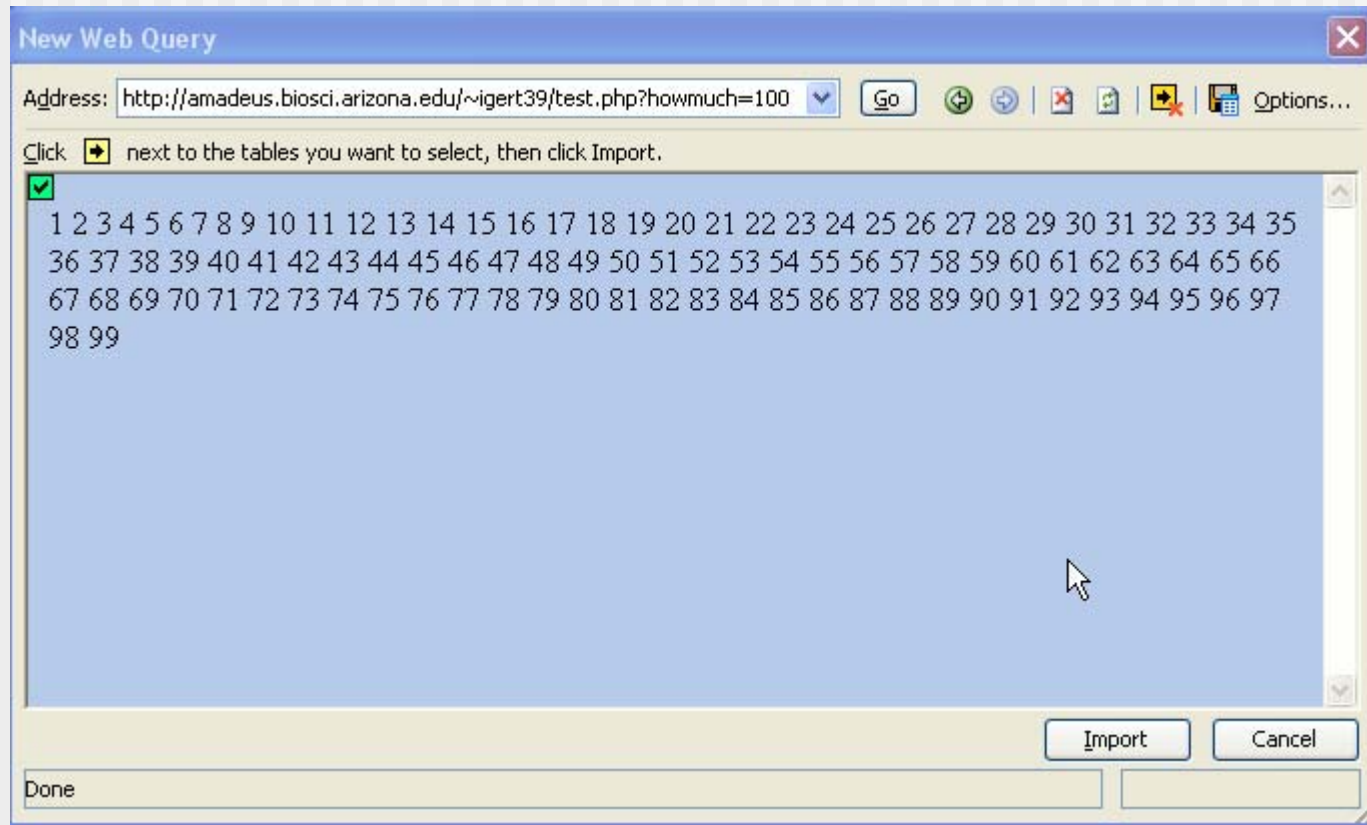
The entire page
 Only the tables
 One or more specific tables on the page.

Enter table name(s) or number(s) separated by commas:

3. Choose how much formatting from the Web page you want your data to keep:

None
 Rich text formatting only
 Full HTML formatting

Excel (ver. 2003)



Change program xl.php

```
<?
for($i=1;$i<=$showmuch;$i++) {
print "$i\tFrom web\n";
}

?>
```

- Save your program and click on the "!" icon in the floating "External Data" window in Excel

Access and MySQL

- We can connect to mysql from Windows using a ODBC driver
- Users can enter data run reports and queries using familiar interface
- Rapid report writing and prototyping

Home Work !

- Due Nov 28th by Midnight
- Write a program to read the file `haplo.csv` located in `/home/student/2005/eeb/hw-3/`
- Create `hw-3` in your home dir.
- Call your program `sieve.pl` (in `hw-3` directory)
- This data is from a prediction program that uses 2 methods (J48 and Support Vector machine) to assign a haplogroup value for a sample (J, R1a etc)
- I would like to find the samples where both methods disagree (i.e 2 diff answers)

Homework (Cont.)

- Input file (`haplo.csv`) has 5 columns separated by ,

You have to:

1. Save all the 5 columns to a file (tab separated values) called (`disagree.txt`) when `J48_method` and `SVM_method` are not same.
2. Count the total number of rows read and total number of rows saved into `disagree.txt` (and answer the questions on the h/w site)

Learning more ...

- <http://uacbt.arizona.edu/>
- Search for Access
- Has PERL, PHP and much more !