CS 50: Introduction to Computer Science
Harvard University
Scribe Notes, Week 10: Wednesday
Tova Wiener

10:07 – 10:11 - Introduction

We gave out a packet with code. In the middle is speller.php. That is a translation of Problem set 5 from C into PHP. Try to notice the differences. While it is almost the same syntactically, the main difference is that variables do not seem to have data types. Different languages are pretty much the same thing but with similar ideas.

This program does run like our C program.

We will now type Require("dictionary.inc"). This corresponds to dictionary.c which you wrote last week.

10:11 – 10:18 Implementing dictionary in PHP

First we will have our global variables, size and dictionary. These have $s before them, and no types.
To declare size we simply write:

```
Function size() {
        Global $size;
        Return $size;
}
```

The line global size just tells php it is a global variable.

Now let's write load.

It will look like this:

```
Function load($dict)
{
        Global $dictionary, $size;
        $fp = fopen($dict, "r");
        $size = 0;
        While((fscanf($fp, "%s", $fp) != 1)
        {
                $dictionary[size] = $word
                $size++;
        }
        Fclose($fp);
}
```

Notice that we do not need to have variable types.

That is it!

Notice we do not need sizes for our arrays.

Now we will implement check.

```
Function check($word)
{
        Global $dictionary;
        Foreach($dictionary as dictword) {
        //This will set up a for loop which will grab one word of dictionary and //putting it
        into dictword
                If($word == $dictword)
                        Return TRUE;
        }
        Return FALSE;
}
```

It works!

10:18 – 10:38 A further understanding of PHP

So why did we learn C?

Now you understand it better. But now we will see that some tasks are much easier when the language handles the difficulty for you.

Our running time in PHP is, however, O(n). We might have to look at all n words for each structure.
We can try to do this with hash tables like you did in C.

We can use words as indices in PHP, as opposed to numbers.

So now we will have load look like this:

```
Function load($dict)
{
        Global $dictionary, $size;
        $fp = fopen($dict, “r”);
        $size = 0;
        While((fscanf($fp, “%s”, $fp) != 1)
        {
                $dictionary[$word] = TRUE;
                $size++;
        }
```

```
        Fclose($fp);
}
```

Now we are storing at the word “foo” the value TRUE. Essentially I am flagging a bit to TRUE for every word in the original dictionary function. So now check can just be:

```
Function check($word)
{
        Global $dictionary;
        If(dictionary[$word] = TRUE) {
                Return TRUE;
        } else
                Return FALSE;
}
```

PHP is not a compiled language but an interpreted language. So you cannot run it directly on the hardware, like a.out. We have installed on nice.fas a program called PHP which is an interpreted. So the long line of code at the top of speller.php says that we need to interpret the code. They are not just 0s and 1s.

PHP-run time

Speller is slower in PHP than in C. This is because PHP is an interpreted language, and it takes time to interpret it. So if we want stuff done fast, we tend to write it in a compiled language like C or Java.

However, coding in PHP is much faster than coding in C.

10:38 – XHTML and CSS

On Monday we left off with XHTML.

Look at css.html. This file will have a link titled “hello, world” which will link to the arbitrary web page hello.com.

We have a line of code which includes styles.css.

Looking at styles.css shows that in the body section, we give colors for the background and text. The a section gives color for links. A:hover says that if you hover over a link it become read, huge and non-underlined.

Table Tag in html allows you to define a table. We declare a table, then start a row with <tr>, and insert data with <td>.

Tables will be useful in laying things out structurally.

(Note, cs50's website also uses css. We can change one line of code to change all of the colors).

On cs50's web page we use tables with borders of 0 to format things nicely.

10:45 – Validation

We can go to a validation website to see if we have valid xhtml code.

10:50- 11:00
There are different forms in HTML.
We will steal some of Google's code and link to it.