

# Software for DHTC Part 2: Interpreted Languages

Christina Koch ([ckoch5@wisc.edu](mailto:ckoch5@wisc.edu))

Research Computing Facilitator

University of Wisconsin - Madison

# Recap

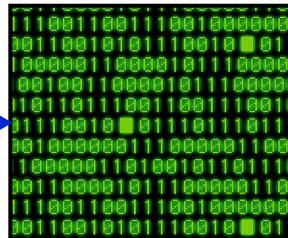
---

- Previous techniques:
  - Compiled code
    - Download compiled binaries
    - Build yourself
  - Wrapper script
    - Run compiled code
    - Use a pre-built installation ← we're going to explore this further in this session

# Interpreted code

- Instead of being compiled and then run...

```
41<?php
42    $f = fopen("login.dat", "r");
43    $i = 0;
44    $c = count($users);
45    $line = $users[$i];
46    if ($line == "") {
47        // User gevonden, password is nu
48        // leeg dus stoppen
49        break;
50    }
51    $pass = $line[1];
52    $pass = trim($pass);
53    if ($pass == "") {
54        // Stop met de 'for'-loop
55        break;
56    }
57    $pass = md5($pass);
58    if ($pass == $password) {
59        return $pass;
60    }
61}
62
63function isLoggedIn() {
64    global $username, $password;
65    if ($username != "" & $password != "") {
66        $pass = md5($password);
67        return ($password == $pass);
68    }
69    return FALSE;
70}
```

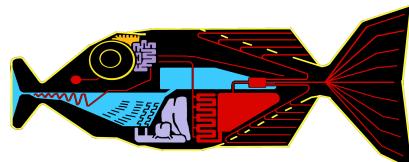


A grid of binary digits (0s and 1s) arranged in a rectangular pattern, representing the interpreted code as binary.



- ...interpreted languages are translated into binary code “on the fly”

```
41<?php
42    $f = fopen("login.dat", "r");
43    $i = 0;
44    $c = count($users);
45    $line = $users[$i];
46    if ($line == "") {
47        // User gevonden, password is nu
48        // leeg dus stoppen
49        break;
50    }
51    $pass = $line[1];
52    $pass = trim($pass);
53    if ($pass == "") {
54        // Stop met de 'for'-loop
55        break;
56    }
57    $pass = md5($pass);
58    if ($pass == $password) {
59        return $pass;
60    }
61}
62
63function isLoggedIn() {
64    global $username, $password;
65    if ($username != "" & $password != "") {
66        $pass = md5($password);
67        return ($password == $pass);
68    }
69    return FALSE;
70}
```





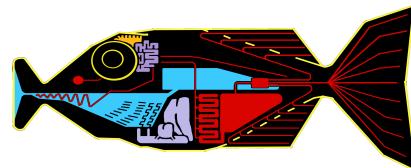
Open Science Grid

# Interpretation

## Script

```
1 $pass = null;
2 // ...
3 // file("login.dat");
4 // ...
5 $i = 0; $i < count($users); $i++
6 $line = $users[$i];
7 if (ereg("~$username(.*)", trim($line))
8 // User gevonden, Password is nu
9 // $pass = $reg[1];
10 break; // Stop met de 'for'-loop
11 }
12 }
13 return $pass;
14 }
15 function isLoggedIn() {
16     global $username, $password;
17     if ($username && $password)
18         $pass = md5(GetPassword());
19     return ($password == $pass) ? TRUE : FALSE;
20 }
```

## Interpreter



uses

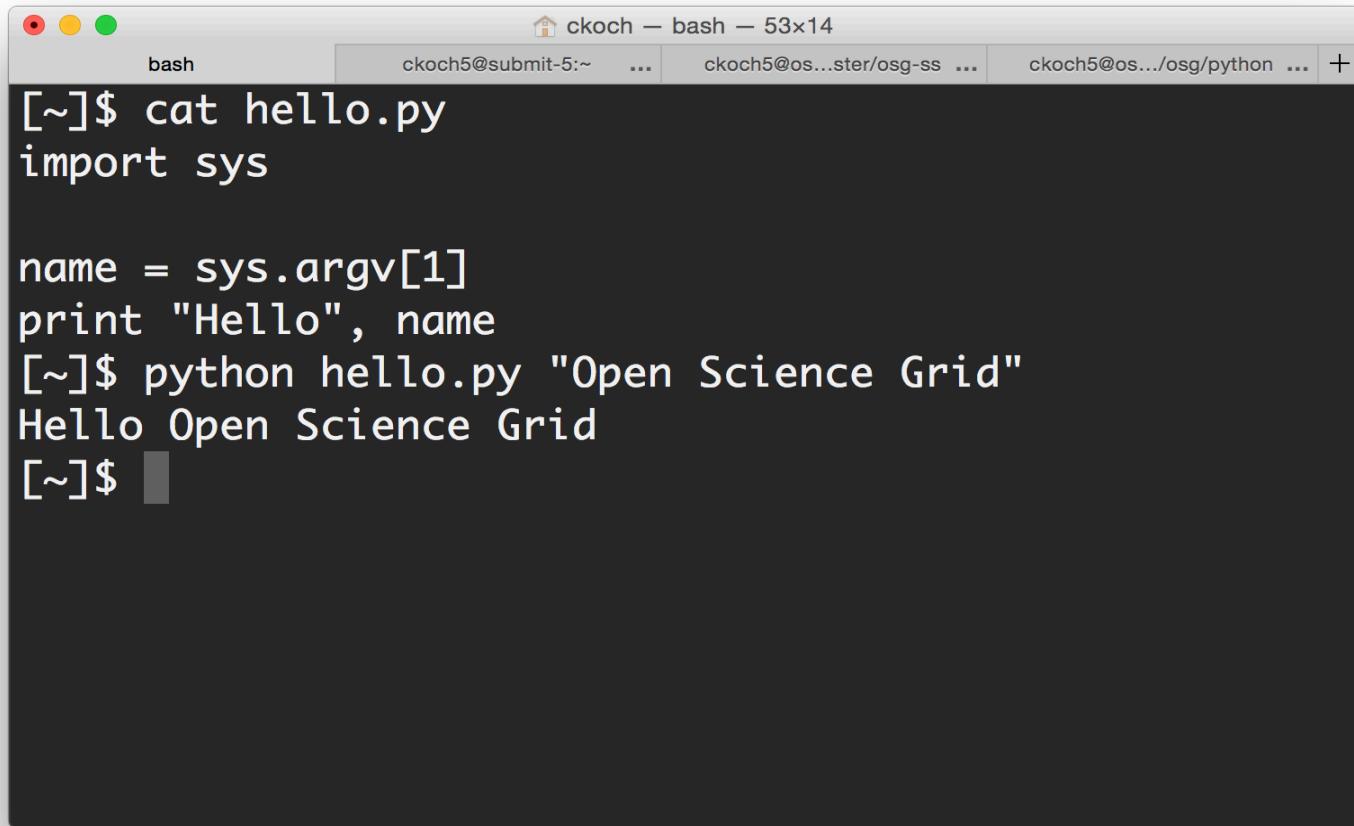


## Libraries

text turns  
into binary  
instructions



# On the command line



A screenshot of a macOS terminal window titled "ckoch — bash — 53x14". The window shows a command-line session:

```
[~]$ cat hello.py
import sys

name = sys.argv[1]
print "Hello", name
[~]$ python hello.py "Open Science Grid"
Hello Open Science Grid
[~]$
```

# Common interpreted languages\*

- Python
- R
- Julia
- Ruby
- Matlab
- Perl
- Javascript



\*Note: the line between interpreted/compiled languages can be fuzzy. Many languages support both options, with one method being more common.

# Running interpreted code in jobs

---

## General procedure

- Need to bring along interpreter and script
- Use a wrapper script as the executable
- Wrapper script will:
  - “Install” the interpreter
  - Run the script using the local installation

# Python on DHTC

1. Create a portable Python installation  
(optional)
2. Bring along:
  - pre-built installation OR Python source code
  - your Python code
3. Use a wrapper script to:
  - unpack pre-built install OR install from source
  - run your Python script

# Exercises

---

- Running Python Jobs
  - Exercise 4.1: Pre-building Python and using that installation
  - Exercise 4.2 (optional): Further Python job customizations
- Work on other exercises from today/yesterday that you weren't able to finish

# Questions?

---

- Now: Hands-on Exercises
  - 3:45 - 5:00pm