# Métodos Computacionales

Sebastian Perez Saaibi

2015.01.28

# Horario de Atención

https://calendly.com/spsaaibi

# Temas

C

- arithmetic operations
- variable types
- print tables / loops
- dynamic memory allocators/pointers
- string arrays
- if / do-while
- functions
- i/o from files

# Notas de Clase: ComputationalMethods

https://github.com/forero/ComputationalMethods/blob/master/notes/C.pdf

# Otras Referencias de C

- A Little C Primer:
  - http://en.wikibooks.org/wiki/A_Little_C_Primer/An_Introductory_C_Program
- *yet another C primer (from a novice, for novices):
  - http://www.vectorsite.net/tscpp.html

# Operaciones Aritméticas

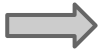
```c
/*
  Different ways to multiply and divide numbers
*/
#include <stdio.h>
int main(void){
  int a;
  int b;
  int c;
  float d;
  float e;
  float f;

  a = 1;
  b = 10;
  c = a/b;

  d = 1.0;
  e = 10.0;
  f = d/e;

  printf("%d %d %d \n", a , b, c);
  printf("%f %f %f \n", d , e, f);
  printf("%d %d %d %f %f %f \n", a,b,c,d,e,f);
return 0; }
```

# Tipos de Variables

- `char` a single byte, holds a character, i.e. letters.
- `int` integers, the range is limited to the capabilities in the host machine, i.e. you cannot count till infinity in a program.
- `float` single-precision floating point, i.e. real numbers.
- `double` double-precision floating point, i.e. real numbers

# Printing

stdin: pointer to the standard input stream.
stout: pointer to the standard output stream.
stderr: pointer to the standard error stream.



In most cases, these are the console (terminal, etc…)

`fprintf` writes formatted text to the output stream you specify.

`printf` is equivalent to writing `fprintf(stdout, ...)` and writes formatted text to wherever the standard output stream is currently pointing.

`sprintf` writes formatted text to an array of `char`, as opposed to a stream.

%d,%s,%f,%e…

How to print a tab?
How to print a %?
How to print dollar amounts in a cashier?

# Keywords

| | | | |
|---|---|---|---|
| auto | double | int | struct |
| break | else | long | switch |
| case | enum | register | typedef |
| char | extern | return | union |
| const | float | short | unsigned |
| continue | for | signed | void |
| default | goto | sizeof | volatile |
| do | if | static | while |

*Table 2.3. Keywords*

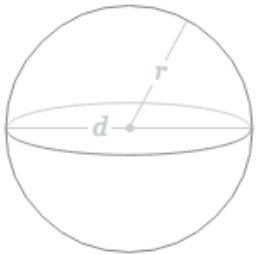
# Tablas/Ciclos

Sphere

Solve for volume

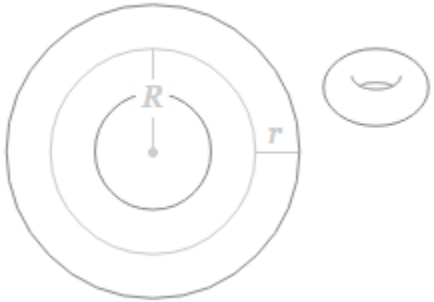
$$V = \frac{4}{3}\pi r^3$$

$r$ Radius Enter value




```c
#include <stdio.h>
#define PI 3.14159

int main(void){
  /*defines the variables*/
  int i;
  float radius;
  float volume;
  float surface;
  /*initialization*/
  radius = 0.0;
  volume = 0.0;
  surface = 0.0;
  printf("Radius Surface Volume\n");
  /*loop over 12 different values for the radius*/
  for(i=0; i<12; i++){
    radius = i;
    surface = 4.0 * PI * radius * radius;
    volume = (4.0 / 3.0 ) * PI * radius * radius * radius;
    /*output the values to the screen*/
    printf("%f %f %f\n", radius, surface, volume);
}
return 0; }
```

# Toroide




Volumen?
Área Superficial?

# Arreglos

```c
/*
  Example of static array definition and the importance of initalization
*/
#include <stdio.h>
int main(void){
  int lista[10]; //define a list of 10 integers
  int i;
  //print the content
  printf("Content before initilization\n");
  for(i=0;i<10;i++){
    printf("%d\n", lista[i]);
  }
  //initialize
  for(i=0;i<10;i++){
    lista[i] = i * 2;
  }
  //print the new content
  printf("Content after initilization\n");
  for(i=0;i<10;i++){
    printf("%d\n", lista[i]);
  }
return 0; }
```