

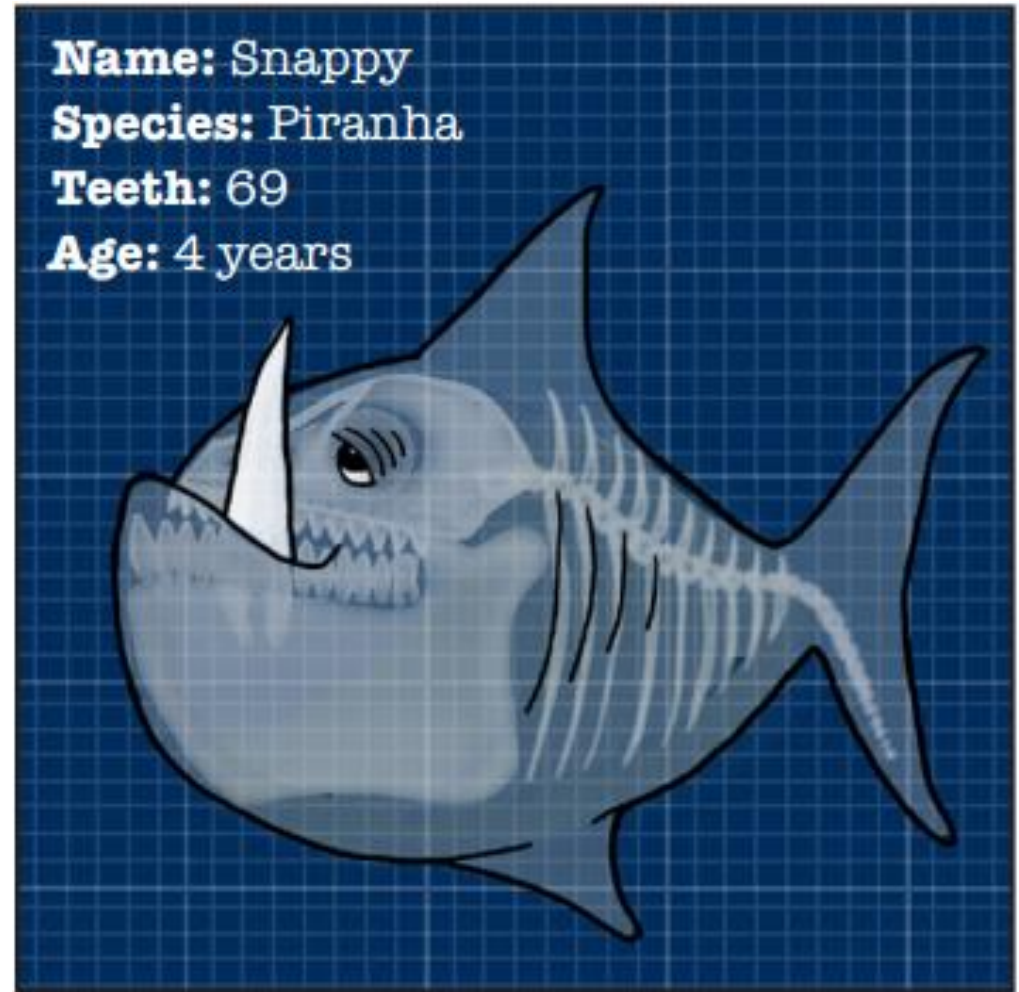
# Técnicas de programación

Gonzalo Noreña

# Structs y typedefs

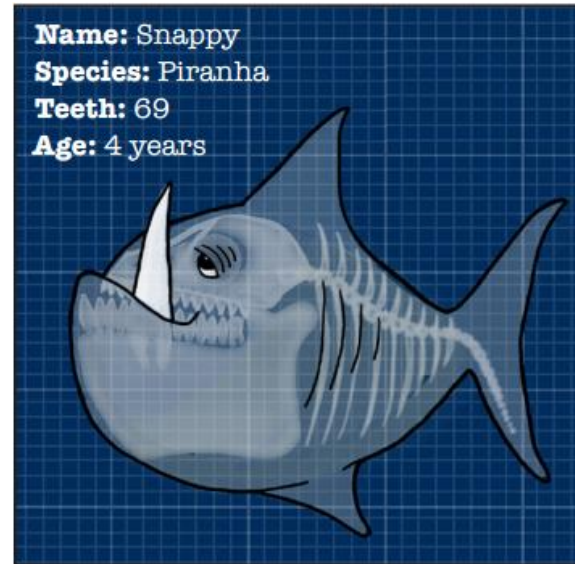
# ¿Por qué surgieron los structs?

A veces es necesario manipular más de un dato a la vez y se desea manipular como un solo elemento



# ¿Cómo se logra este comportamiento?

```
struct Fish {  
    char nombre[25];  
    char especie[20];  
    int numDientes;  
    int edad;  
};
```



## Structs ( structured data type)

Son conjuntos de variables relacionadas bajo un mismo nombre que se “envuelven” en un nuevo tipo de datos.

# ¿Cuáles son las reglas de sintaxis a considerar?

```
struct Fish {  
    char nombre[25];  
    char especie[20];  
    int numDientes;  
    int edad;  
};
```

Se definen con la palabra reservada **struct**.  
**Llaves al inicio y final. Punto y coma para**  
**terminar**

# ¿Cómo se usan los structs?

```
struct Fish {  
    char nombre[25];  
    char especie[20];  
    int numDientes;  
    int edad;
```



Fin de estructura

Etiqueta de  
estructura

Miembros de la  
estructura

Crea un nuevo tipo de datos  
llamado *struct Fish*

Los miembros pueden  
ser de cualquier tipo

Crea una variable de tipo struct  
Empleado

```
#include <stdio.h>  
#include "Structs.h"  
int main()  
{  
    struct Fish piranita= {"Snappy", "Piranha", 69, 4};  
    return 0;  
}
```



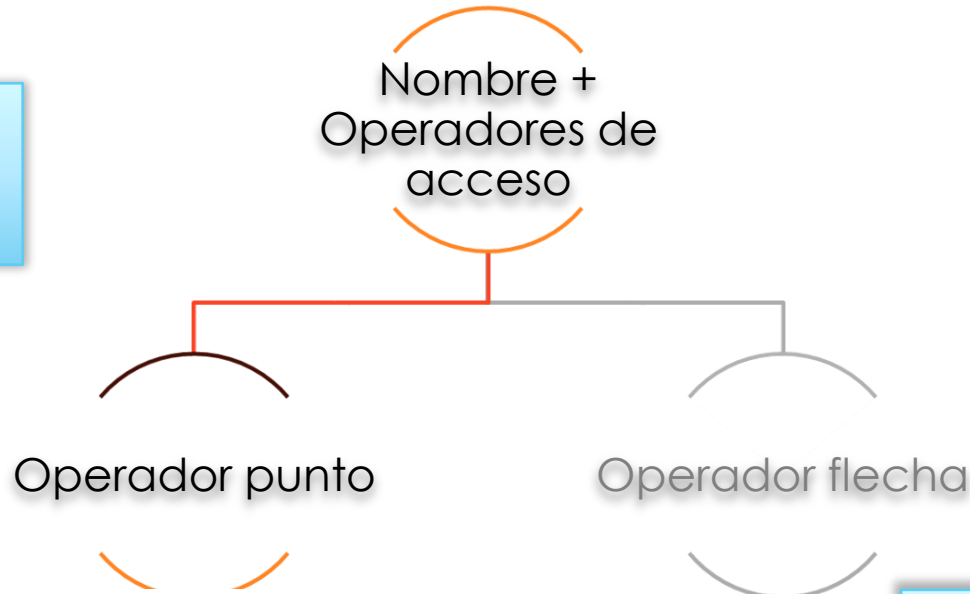
# ¿Qué tipo de variables se pueden definir con un struct?

```
struct Fish piranita = {"Snappy", "Piranha", 69, 4};  
struct Fish arregloPiranitas[5];  
struct Fish *pStructPiranita = NULL;  
struct Fish piranitaDos;
```

**!Variables de cualquier tipo! : arreglos, apuntadores, variables sencillas.**

# ¿Cómo se accede a los miembros de los structs?

Se tiene el objeto



```
struct Fish fish;
printf("ingrese un nombre para el"  
      "pez max 24 caracteres\n");  
fflush(stdin);  
fgets(fish.nombre, 25, stdin);
```

Se tiene el  
apuntador al  
objeto  
**Ver los  
siguientes  
slides**



# Ejemplo Structs con memoria estática.

```
struct Fish llenarFishMemEstatica(){  
  
    struct Fish fish;  
    printf("ingrese un nombre para el pez max 24 caracteres\n");  
    fflush(stdin);  
    fgets(fish.nombre,25,stdin);  
    printf("ingrese la especie para el pez max 24 caracteres\n");  
    fflush(stdin);  
    fgets(fish.especie,20,stdin);  
    printf("ingrese el numero de dientes \n");  
    scanf("%d", &fish.numDientes);  
    printf("ingrese la edad del pez\n");  
    scanf("%d", &fish.edad);  
  
    return fish;  
}
```

Crea una estructura y devuelve una copia al salir de la función

```
struct Fish piranitaDos;  
piranitaDos=llenarFishMemEstatica();
```

# Sobre structs revisando

Asignar valores a un struct:

**NombreStruct nomVariable= {lista  
valores separadas por comas};**

Acceder a valores del struct:

**nombreVariable (punto) nombre  
variable del struct;**

Ejm fish.numDientes

# Typedef y structs

Facilita la definición de struct al darle un **alias** a la definición.

En el .h

```
struct Fish {  
    char nombre[25];  
    char especie[20];  
    int numDientes;  
    int edad;  
};
```

```
typedef struct Fish fish_t;
```

Para definir una variable

```
/*Con typedef*/  
fish_t llenarFishTypeDefStatic();
```

```
struct Fish llenarFishTypeDefStatic(){  
    /* La lógica es igual, la diferencia es que  
    la declaración se la variable se hace con fish_t*/  
    fish_t fish;  
    printf("ingrese un nombre para"  
        " el pez max 24 caracteres\n");  
    fflush(stdin);  
    fgets(fish.nombre,25,stdin);
```

# Typedef y struct en una sola línea

```
typedef struct Fish2 {  
    char nombre[25];  
    char especie[20];  
    int numDientes;  
    int edad;  
} fish_t2;
```

# Actividad en clase

- Cree un struct con datos alusivos a una producción de Netflix **(Nombre, Género, año de producción y tipo(serie o película))**
- Cree un arreglo de producciones, de manera que puedan ingresarse 5 producciones dentro de su arreglo.
- Cree una función que le permita llenar los datos de una producción y retorne la producción lista para ser agregada en su arreglo.
- Similar al punto anterior (llenar un struc), pero ahora cree un procedimiento que recibe como parámetro por referencia un struct (un elemento de su arreglo, por ejemplo), de manera que los cambios sean persistentes.
- Cree un procedimiento, donde el usuario pueda llenar los datos de todas las producciones.
- Cree un procedimiento que imprima el nombre de sólo aquellas producciones que fueron realizadas entre el año 2010 y 2015

# Referencias

[1] Harvey M. Deitel, Paul J. Deitel, C how to program in C/C++. 6th edition. 2010.

[2] Head First C. David Griffiths, Dawn Griffiths. O'Reilly. 2012. Url <http://www.it-ebooks.info/book/704/>