Void pointers

void pointers (void *)

Allow us to implement **generic** functions: work with many types.
Also used for dealing with **buffers of bytes**. (malloc, free, ...)
Polymorphism - works with many types.

- Pointers are the same size regardless of type
    - 4 bytes -- 32 bit   8 bytes -- 64 bit
    - *sizeof(int *) == sizeof(double *) == sizeof(country_t *) == sizeof(void *).*

- Dereferencing a void pointer is illegal!   Must cast to some real type.

- sizeof(void) is undefined. [sizeof(*vptr) == undefined for void *vptr]

void pointers (void *)

void *malloc(size_t size);

... returns a void *

```
    // implicit cast
    int *arr = malloc(5 * sizeof(int));

    // explicit cast
    int *arr = (int *) malloc(5 * sizeof(int));

    // recommended: if you are declaring, then initialising
    int *arr;

    arr = (int *) malloc(5 * sizeof(int));
```

void pointers (void *)

Why not just do...
```
void *arr = malloc(5 * sizeof(int));
```

void pointers (void *)

Why not just do
    void *arr = malloc(5 * sizeof(int));

Void pointer issues
- Cannot dereference void pointers
    - arr has type void *
    - *arr has type void   … ??
- sizeof(void) is undefined
    - arr[i] == *(arr + sizeof(*arr) * i)
    - but sizeof(*arr) == sizeof(void) == undefined
    - arr[i] is undefined.
- Would need to use casts everywhere.
- Nothing protecting you from casting back to the wrong type
    - ...

void pointers (void *)

Why not just do
    void *arr = malloc(5 * sizeof(int));

Void pointer issues
 - ...
 - Nothing protecting you from casting back to the wrong type

```
int x = 2;     // x is 4 bytes
int *p = &x;   // p is 8 bytes
void *p1 = p; // p1 is 8 bytes
double *p2 = p1;

sizeof(*p) = ??
sizeof(*p1) = ??
sizeof(*p2) = ??
```

void pointers (void *)

Why not just do
    void *arr = malloc(5 * sizeof(int));

Void pointer issues
 - ...
 - Nothing protecting you from casting back to the wrong type

```
int x = 2;      // x is 4 bytes
int *p = &x;    // p is 8 bytes
void *p1 = p;   // p1 is 8 bytes
double *p2 = p1;
```

```
sizeof(*p) = 4
sizeof(*p1) = undefined!
sizeof(*p2) = 8
```