Arrays and loops

Adapted from materials by Dr. Carrier

Tim Wildsmith - Unsplash

Basic arrays

Basic arrays

All elements in a C array have the same type

Basic arrays

All elements in a C array have the same type

For now, we are talking about *static* arrays

- I.e., fixed size

Basic arrays

All elements in a C array have the same type

For now, we are talking about *static* arrays

- I.e., fixed size

Declaring:

Basic arrays

All elements in a C array have the same type

For now, we are talking about *static* arrays

- I.e., fixed size

Declaring:

```
int arr[10];
```

Basic arrays

All elements in a C array have the same type

For now, we are talking about *static* arrays

- I.e., fixed size

Declaring:

```
int arr[10];
```

```
int arr[3] = {2, 4, 6};
```

Basic arrays

All elements in a C array have the same type

For now, we are talking about *static* arrays

- I.e., fixed size

Declaring:

```
int arr[10];
```

```
int arr[3] = {2, 4, 6};
```

```
char arr[5];
```

Basic arrays

All elements in a C array have the same type

For now, we are talking about *static* arrays

I.e., fixed size

Declaring:

```
int arr[10];
```

```
int arr[3] = {2, 4, 6};
```

```
char arr[5];
```

```
char arr[] = “hello world”;
```

Array indexing

We still access elements like in every other language (zero-indexed)

```
int arr[3] = {2, 4, 6};
```

Array indexing

We still access elements like in every other language (zero-indexed)

```
int arr[3] = {2, 4, 6};
```

```
int x = arr[1];
```

Array indexing

We still access elements like in every other language (zero-indexed)

```
int arr[3] = {2, 4, 6};
```

```
int x = arr[1];
```

```
arr[2] = 0;
```

For loops

For loops

```
unsigned int i;
for(i = 0; i < 10; i++){

...
}
```

For loops

```
unsigned int i;
for(i = 0; i < 10; i++){

...
}
```

Newer versions of C allow you to create iterative variable inside the loop line

While loops

```
while(condition){

...
}
```

While loops

```
while(condition){

...
}
```

```
int i = 10;

while(i >= 0){

...

i--;
}
```