Function

Define function

```
return_type function_name (parameters) {
        statements
        return value;
}
```

name: what you call the function
parameters: information passed into function
return: information given back from the function

Anatomy of a function

```
int sum(int a, int b) {
    int sum = a + b;
    return sum;
}
```

Function Definition

```
int main() {
    int z;
    z = sum(5, 4);
    printf("%d", z);
    return 0;
}
```

Function Call

Anatomy of a function

Name

```
int sum(int a, int b) {
    int sum = a + b;
    return sum;
}

int main() {
    int z;
    z = sum(5, 4);
    printf("%d", z);
    return 0;
}
```

Anatomy of a function

Inputs expected

```
int sum(int a, int b) {
    int sum = a + b;
    return sum;
}

int main() {
    int z;
    z = sum(5, 4);
    printf("%d", z);
    return 0;
}
```

Inputs given

Anatomy of a function

parameters

```
int sum(int a, int b) {
    int sum = a + b;
    return sum;
}

int main() {
    int z;
    z = sum(5, 4);
    printf("%d", z);
    return 0;
}
```

argument

Anatomy of a function

```
int sum(int a, int b) {
    int sum = a + b;
    return sum;
}

int main() {
    int z;
    z = sum(5, 4);
    printf("%d", z);
    return 0;
}
```

Body of a function

Anatomy of a function

```
int sum(int a, int b) {
    int sum = a + b;
    return sum;
}

int main() {
    int z;
    z = sum(5, 4);
    printf("%d", z);
    return 0;
}
```

Ends the function and returns back a value

This call evaluates to the value returned

Anatomy of a function

```
int sum(int a, int b) {
    int sum = a + b;
    return sum;
}
```

```
int main() {
    int z;
    z = sum(5, 4);
    printf("%d", z);
    return 0;
}
```

A function also

Anatomy of a function

Main memory

```
int sum(int a, int b) {
    int sum = a + b;
    return sum;
}
int main() {
    int z = sum(5, 4);
    printf("%d", z);
    return 0;
}
```

Terminal

Anatomy of a function

Main memory

```
int sum(int a, int b) {
    int sum = a + b;
    return sum;
}
int main() {
    int z = sum(5, 4);
    printf("%d", z);
    return 0;
}
```

Terminal

Anatomy of a function

Sum memory

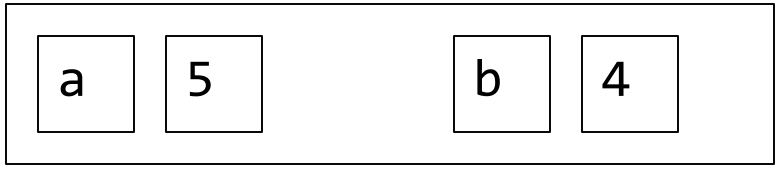
Main memory

```
int sum(int a, int b) {
    int sum = a + b;
    return sum;
}
int main() {
    int z = sum(5, 4);
    printf("%d", z);
    return 0;
}
```

Terminal

Anatomy of a function

Sum memory



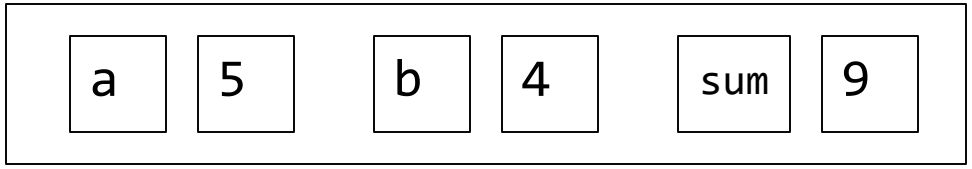
Main memory

```
int sum(int a, int b) {
    int sum = a + b;
    return sum;
}
int main() {
    int z = sum(5, 4);
    printf("%d", z);
    return 0;
}
```

Terminal

Anatomy of a function

Sum memory



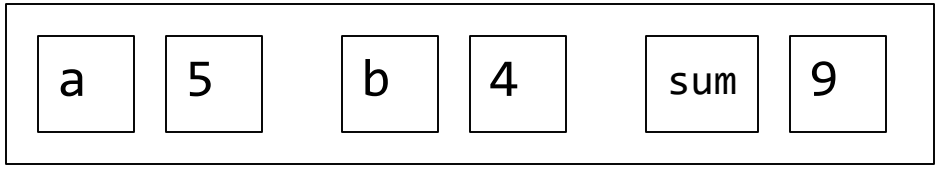
Main memory

```
int sum(int a, int b) {
    int sum = a + b;
    return sum;
}
int main() {
    int z = sum(5, 4);
    printf("%d", z);
    return 0;
}
```

Terminal

Anatomy of a function

Sum memory



Main memory

```
int sum(int a, int b) {
    int sum = a + b;
    return sum;
}
int main() {
    int z = sum(5, 4);
    printf("%d", z);
    return 0;
}
```

9

Terminal

Anatomy of a function

Main memory


```
int sum(int a, int b) {
    int sum = a + b;
    return sum;
}
int main() {
    int z = sum(5, 4);
    printf("%d", z);
    return 0;
}
```

9

Terminal

Anatomy of a function

Main memory


```
int sum(int a, int b) {
    int sum = a + b;
    return sum;
}
int main() {
    int z = sum(5, 4);
    printf("%d", z);
    return 0;
}
```

9

Terminal

Anatomy of a function

```
int sum(int a, int b) {
    int sum = a + b;
    return sum;
}
int main() {
    int z = sum(5, 4);
    printf("%d", z);
    return 0;
}
```

Main memory

Terminal

```
9
```

Anatomy of a function

Main memory


```
int sum(int a, int b) {
    int sum = a + b;
    return sum;
}
int main() {
    int z = sum(5, 4);
    printf("%d", z);
    return 0;
}
```

```
Terminal
9
```

Anatomy of a function

Main memory

```
int sum(int a, int b) {
    print(“%d”, a + b)
}
int main() {
    sum(5, 4);
    return 0;
}
```

Terminal

Anatomy of a function

Sum memory

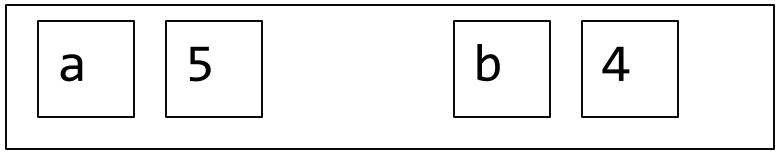
Main memory

```
int sum(int a, int b) {
    print("%d", a + b)
}
int main() {
    sum(5, 4);
    return 0;
}
```

Terminal

Anatomy of a function

Sum memory



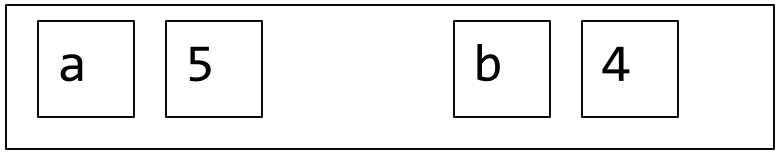
Main memory

```
int sum(int a, int b) {
    print("%d", a + b)
}
int main() {
    sum(5, 4);
    return 0;
}
```

Terminal

Anatomy of a function

Sum memory



Main memory

```
int sum(int a, int b) {
    print("%d", a + b)
}
int main() {
    sum(5, 4);
    return 0;
}
```

```
Terminal
9
```

Anatomy of a function

Main memory

```
int sum(int a, int b) {
    print(“%d”, a + b)
}
int main() {
    sum(5, 4);
    return 0;
}
```

Terminal
9

Anatomy of a function

Main memory

```
int sum(int a, int b) {
    print(“%d”, a + b)
}
int main() {
    sum(5, 4);
    return 0;
}
```

Terminal
9

Function Prototype

```
int sum(int a, int b);

int main() {
    sum(5, 4);
    return 0;
}

int sum(int a, int b) {
    print(“%d”, a + b)
}
```

Function Prototype

```
int sum(int, int);

int main() {
    sum(5, 4);
    return 0;
}

int sum(int a, int b) {
    print(“%d”, a + b)
}
```

Passing arrays to a Function

```
double average(int a[], int n) {

}

int main() {
    int x[5] = {1, 2, 3, 4, 5};
    average(x, N);
}
```

Passing arrays to a Function

```
double average(int a[][5], int r, int c) {

}

int main() {
    int x[2][5] = {1, 2, 3, 4, 5,
      6, 0, 1, , 4, 6};
    average(x, 2, 5);
}
```