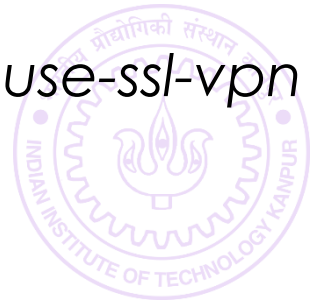
The Eloquent Mr C

ESC101: Foundations of Computing

Purushottam Kar

Announcements

- CHM101 sectioning has caused some clashes
  - This week we are allowing students to attend lab out of order
  - Met DoAA officials to resolve the matter – lets hope for the best ☺
- Unable to access the website?
  - Are you using a data plan on your smart phone?
  - Our course website, Prutor are *internal*, not accessible outside IITK
    - *Solution 1: use IITK computers (CC, NCL, hostel)*
    - *Solution 2: install a VPN app on your smart phone https://www.iitk.ac.in/ccnew/index.php/13-network/99-how-to-use-ssl-vpn*
  - Gradescope, Piazza are accessible from all places
  - Thanks to Nikita Awasthi (tutor) for pointing this out



Writing pretty code is an art

Writing pretty code is an art

Very important in industry – large groups collaborate

Writing pretty code is an art

Very important in industry – large groups collaborate

Important even for solo projects – maintenance

Writing pretty code is an art

Very important in industry – large groups collaborate

Important even for solo projects – maintenance

Will learn several good coding habits over time

Writing pretty code is an art

Very important in industry – large groups collaborate

Important even for solo projects – maintenance

Will learn several good coding habits over time

- Commenting, Indentation, Code-structuring

Writing pretty code is an art

Very important in industry – large groups collaborate

Important even for solo projects – maintenance

Will learn several good coding habits over time

Commenting, Indentation, Code-structuring

```
int main(){
int a;
int b;
a = 5;
b = 4;
a + b;
return 0;
}
```

Writing pretty code is an art

Very important in industry – large groups collaborate

Important even for solo projects – maintenance

Will learn several good coding habits over time

Commenting, Indentation, Code-structuring

```
int main(){
int a;
int b;
a = 5;
b = 4;
a + b;
return 0;
}
```

```
int main(){
    int a;
    int b;
    a = 5;
    b = 4;
    a + b;
    return 0;
}
```

Writing pretty code is an art

Very important in industry – large groups collaborate

Important even for solo projects – maintenance

Will learn several good coding habits over time

Commenting, Indentation, Code-structuring

```
int main(){
int a;
int b;
a = 5;
b = 4;
a + b;
return 0;
}
```

```
int main(){
    int a;
    int b;
    a = 5;
    b = 4;
    a + b;
    return 0;
}
```

```
int main(){
int a;
            int b;
    a=5;
        b=   4;
  a+b;
return 0;
    }
```

Writing pretty code is an art

Very important in industry – large groups collaborate

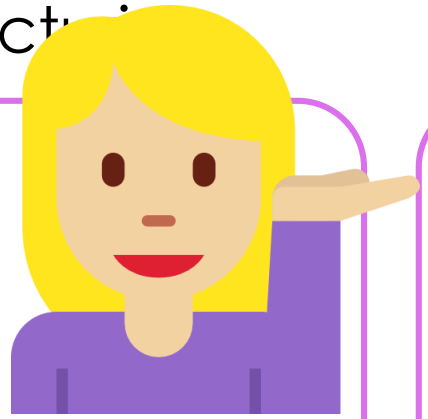
Important even for solo projects – maintenance

Will learn several good coding habits over time

Commenting, Indentation, Code-struct

```
int main(){
int a;
int b;
a = 5;
b = 4;
a + b;
return 0;
}
```

```
int main(){
    int a;
    int b;
    a = 5;
    b = 4;
    a + b;
    return 0;
}
```



```
int main(){
int a;
            int b;
    a=5;
        b=   4;
  a+b;
return 0;
    }
```

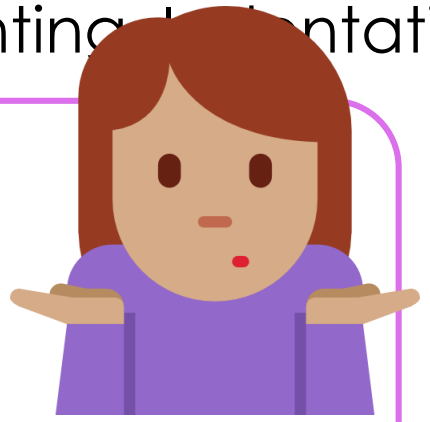
Writing pretty code is an art

Very important in industry – large groups collaborate

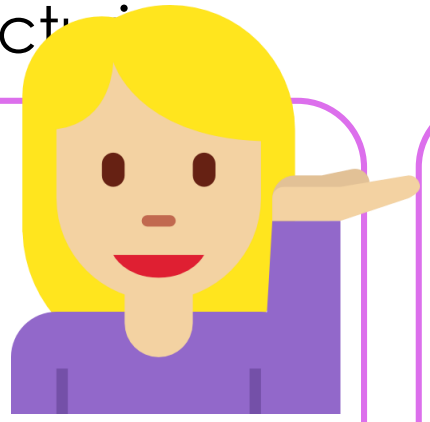
Important even for solo projects – maintenance

Will learn several good coding habits over time

Commenting, Indentation, Code-structuring



```
int main(){
int a;
int b;
a = 5;
b = 4;
a + b;
return 0;
}
```



```
int main(){
    int a;
    int b;
    a = 5;
    b = 4;
    a + b;
    return 0;
}
```

```
int main(){
int a;
            int b;
    a=5;
        b=   4;
  a+b;
return 0;
    }
```

Writing pretty code is an art

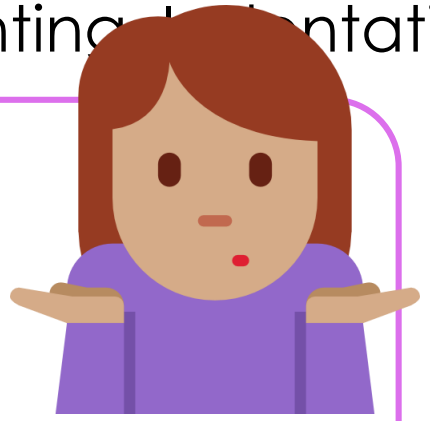
Very important in industry – large groups collaborate

Important even for solo projects – maintenance

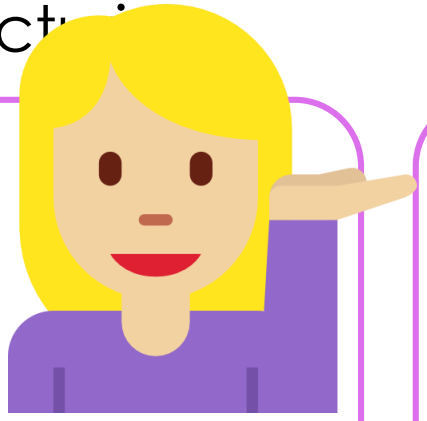
Will learn several good coding habits over time

Commenting, Indentation, Code-structuring

```
int main(){
int a;
int b;
a = 5;
b = 4;
a + b;
return 0;
}
```



```
int main(){
    int a;
    int b;
    a = 5;
    b = 4;
    a + b;
    return 0;
}
```



```
int main(){
int a;
            int b;
   a=5;
       b=   4;
  a+b;
return 0;
   }
```

A handy shorthand

HOW WE MUST SPEAK TO MR. COMPILER

HOW WE USUALLY SPEAK TO A HUMAN

A handy shorthand

HOW WE MUST SPEAK TO MR. COMPILER

HOW WE USUALLY SPEAK TO A HUMAN

Hello

a is a variable.

b is another variable.

a = 5.

b = 4.

Please add them.

Goodbye

A handy shorthand

HOW WE MUST SPEAK TO MR. COMPILER

```
int main(){
int a;
int b;
a = 5;
b = 4;
a + b;
return 0;
}
```

HOW WE USUALLY SPEAK TO A HUMAN

Hello
a is a variable.
b is another variable.
a = 5.
b = 4.
Please add them.
Goodbye

A handy shorthand

HOW WE MUST SPEAK TO MR. COMPILER

```
int main(){
    int a, b;
    a = 5;
    b = 4;
    a + b;
    return 0;
}
```

HOW WE USUALLY SPEAK TO A HUMAN

Hello

a and b are two variables.

a = 5.

b = 4.

Please add them.

Goodbye

A handy shorthand

HOW WE MUST SPEAK TO MR. COMPILER

```
int main(){
    int a, b;
    a = 5, b = 4;
    a + b;
    return 0;
}
```

HOW WE USUALLY SPEAK TO A HUMAN

Hello

a and b are two variables.

a = 5 and b = 4.

Please add them.

Goodbye

A handy shorthand

HOW WE MUST SPEAK TO MR. COMPILER

```
int main(){
    int a = 5, b = 4;
    a + b;
    return 0;
}
```

HOW WE USUALLY SPEAK TO A HUMAN

Hello

a = 5 and b = 4 are two variables.

Please add them.

Goodbye

A handy shorthand

HOW WE MUST SPEAK TO MR. COMPILER

```
int main(){
    int a = 5, b = 4;
    a + b;
    return 0;
}
```

HOW WE USUALLY SPEAK TO A HUMAN

Hello

a = 5 and b = 4 are two variables.

Please add them.

Goodbye

A handy shorthand

HOW WE MUST SPEAK TO MR. COMPILER

```
int main(){
    int a = 5, b = 4;
    a + b;
    return 0;
}
```

HOW WE USUALLY SPEAK TO A HUMAN

Hello

a = 5 and b = 4 are two variables.

Please add them.

Goodbye

A handy shorthand

HOW WE MUST SPEAK TO MR. COMPILER

```
int main(){
    int a = 5, b = 4;
    a + b;
    return 0;
}
```

HOW WE USUALLY SPEAK TO A HUMAN

Hello

a = 5 and b = 4 are two vari

Plea em.

Goo

How will I remember all this?

A handy shorthand

HOW WE MUST SPEAK TO MR. COMPILER

```
int main(){
    int a = 5, b = 4;
    a + b;
    return 0;
}
```

HOW WE USUALLY SPEAK TO A HUMAN

Hello

a = 5 and b = 4 are two vari

Plea em.

Goo

How will I remember all this?

A handy shorthand

HOW WE MUST SPEAK TO MR. COMPILER

```
int main()
    int a =
    a + b;
    return
}
```

Explore, practice. It will take only few days to internalize. No need to mug!

HOW WE USUALLY SPEAK TO A HUMAN

Hello

a = 5 and b = 4 are two vari

Plea m.

Goo

How will I remember all this?

Printing the sum of two numbers

Printing the sum of two numbers

HOW WE USUALLY SPEAK TO A HUMAN

Printing the sum of two numbers

HOW WE USUALLY SPEAK TO A HUMAN

Printing the sum of two numbers

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Printing the sum of two numbers

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

Printing the sum of two numbers

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

Printing the sum of two numbers

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Printing the sum of two numbers

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

Printing the sum of two numbers

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

Please tell me value of c.

Printing the sum of two numbers

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

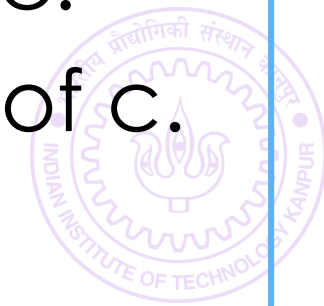
a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

Please tell me value of c.

Goodbye



Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

Please tell me value of c.

Goodbye

Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

Please tell me value of c.

Goodbye

Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

Please tell me value of c.

Goodbye

Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

Please tell me value of c.

Goodbye

Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
int a, b, c;
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

Please tell me value of c.

Goodbye

Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
int a, b, c;
a = 5, b = 4;
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

Please tell me value of c.

Goodbye

Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
int a, b, c;
a = 5, b = 4;
c = a + b;
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

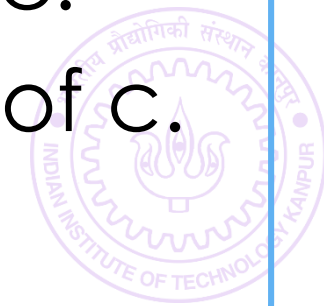
a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

Please tell me value of c.

Goodbye



Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
int a, b, c;
a = 5, b = 4;
c = a + b;
printf("%d",c);
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

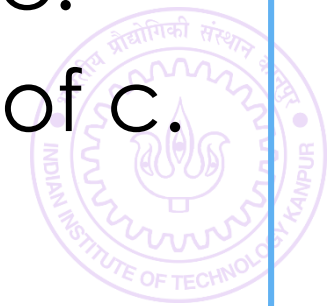
a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

Please tell me value of c.

Goodbye



Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
int a, b, c;
a = 5, b = 4;
c = a + b;
printf("%d",c);
return 0;
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

Please tell me value of c.

Goodbye

Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
int a, b, c;
a = 5, b = 4;
c = a + b;
printf("%d",c);
return 0;
}
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

Please tell me value of c.

Goodbye

Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
int a, b, c;
a = 5, b = 4;
c = a + b;
printf("%d",c);
return 0;
}
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

Please tell me value of c.

Goodbye

Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
int a, b, c;
a = 5, b = 4;
c = a + b;
printf("%d",c);
return 0;
}
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

Please tell me value of c.

Goodbye

Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
int a, b, c;
a = 5, b = 4;
c = a + b;
printf("%d",c);
return 0;
}
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

Please tell me value of c.

Goodbye

Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
int a, b, c;
a = 5, b = 4;
c = a + b;
printf("%d",c);
return 0;
}
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

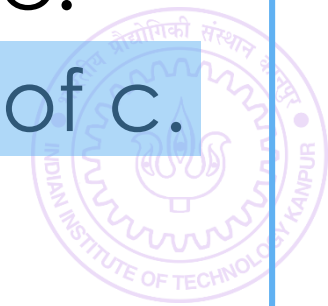
a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

Please tell me value of c.

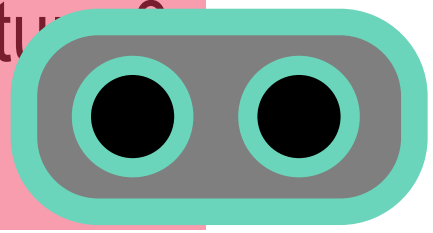
Goodbye



Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
int a, b, c;
a = 5, b = 4;
c = a + b;
printf("%d",c);
retu
}
```



HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

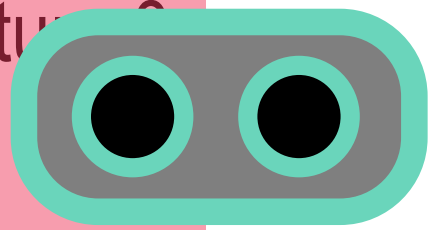
Please tell me value of c.

Goodbye

Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
int a, b, c;
a = 5, b = 4;
c = a + b;
printf("%d",c);
retu
}
```



a

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

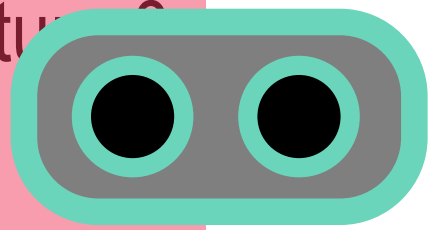
Please tell me value of c.

Goodbye

Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
int a, b, c;
a = 5, b = 4;
c = a + b;
printf("%d",c);
retu
}
```



a b

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

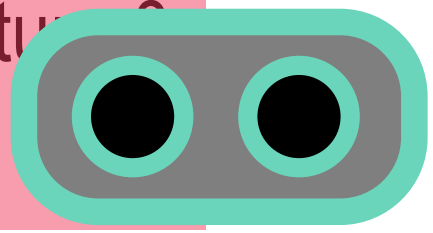
Please tell me value of c.

Goodbye

Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
int a, b, c;
a = 5, b = 4;
c = a + b;
printf("%d",c);
retu
}
```



a

b

c

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

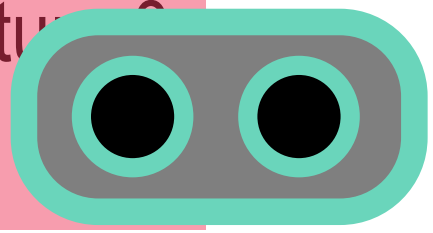
Please tell me value of c.

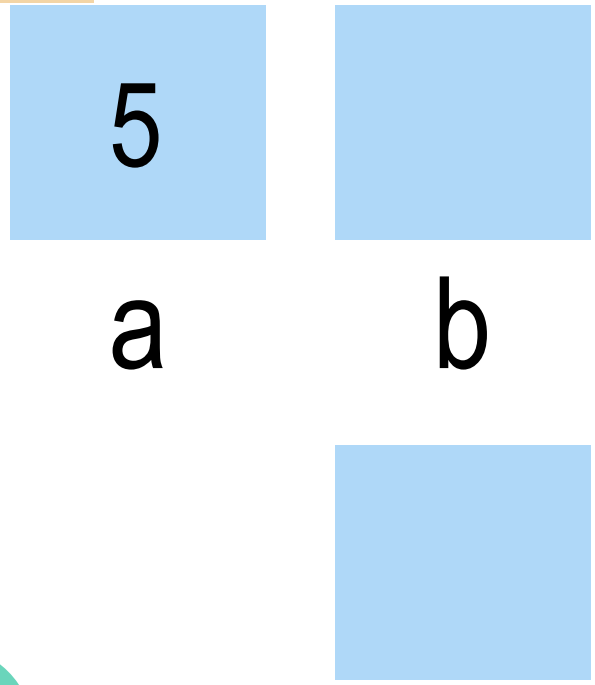
Goodbye

Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
int a, b, c;
a = 5, b = 4;
c = a + b;
printf("%d",c);
retu
}
```





HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

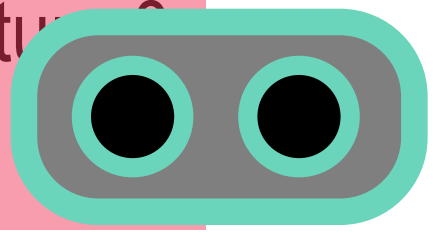
Please tell me value of c.

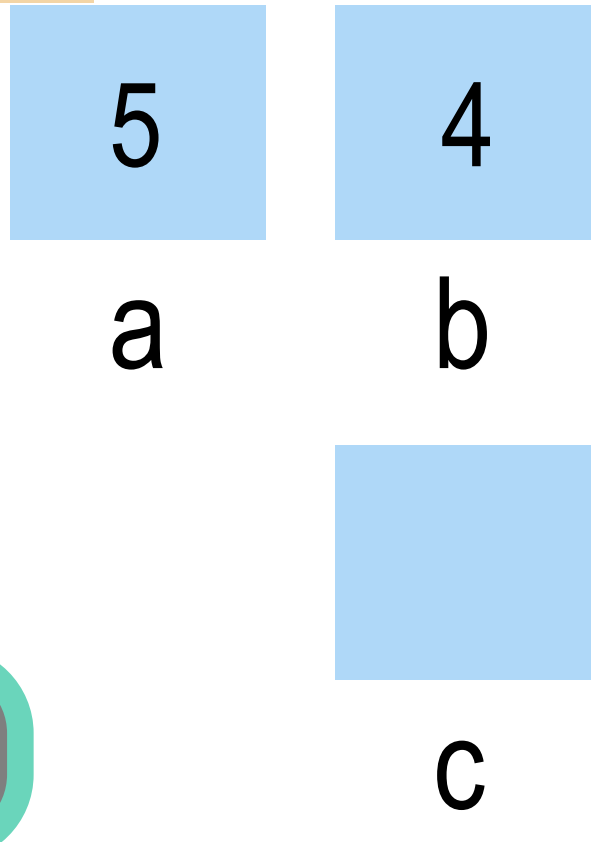
Goodbye

Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
int a, b, c;
a = 5, b = 4;
c = a + b;
printf("%d",c);
retu
}
```





HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

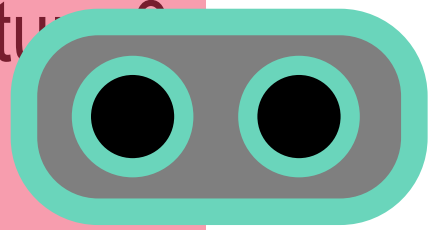
Please tell me value of c.

Goodbye

Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
int a, b, c;
a = 5, b = 4;
c = a + b;
printf("%d",c);
retu
}
```



5	4
a	b

9
c

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

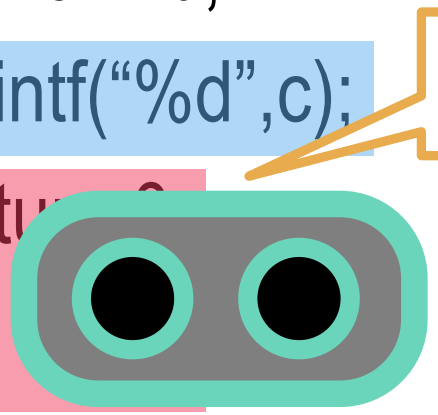
Please tell me value of c.

Goodbye

Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
int a, b, c;
a = 5, b = 4;
c = a + b;
printf("%d",c);
retu
}
```



5	4
a	b

9

9
c

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c.

Please tell me value of c.

Goodbye

Printing the sum of two numbers

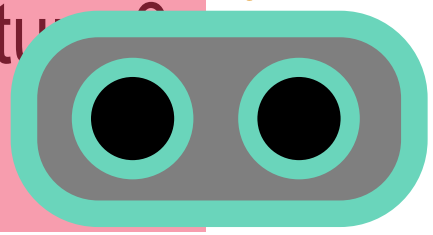
HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
int a, b, c;
a = 5, b = 4;
c = a + b;
printf("%d",c);
retu
}
```

5	4
a	b

9
c

9



HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

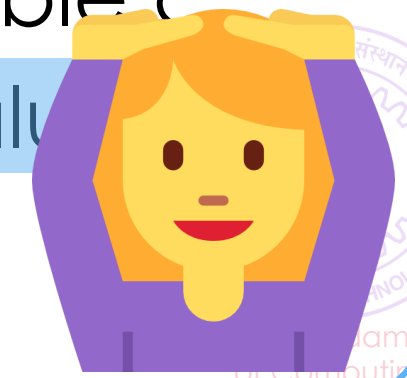
a,b,c are variables.

a = 5 and b = 4.

Please add them and put the result in variable c

Please tell me valu

Goodbye

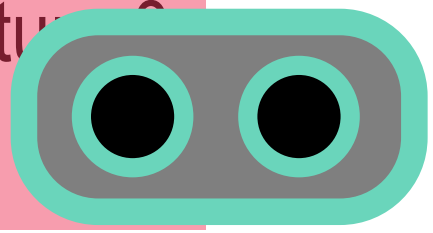


Printing the sum of two numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include<stdio.h>
int main(){
int a, b, c;
a = 5, b = 4;
c = a + b;
printf("%d",c);
retu
}
```

5	4
a	b
9	9
	c



HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

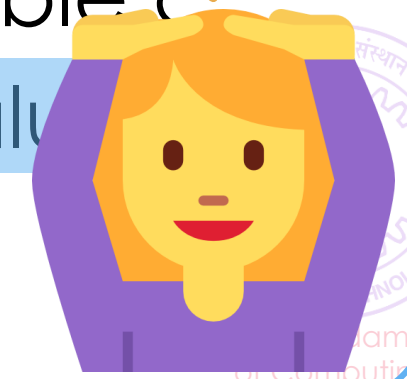
Mr C is happy now. He did not have to throw away the value 9 ☺

s.

d put the result in variable c

Please tell me valu

Goodbye



The true power of printf

The true power of printf

We have seen how to make Mr. C

The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”

The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable

The true power of printf

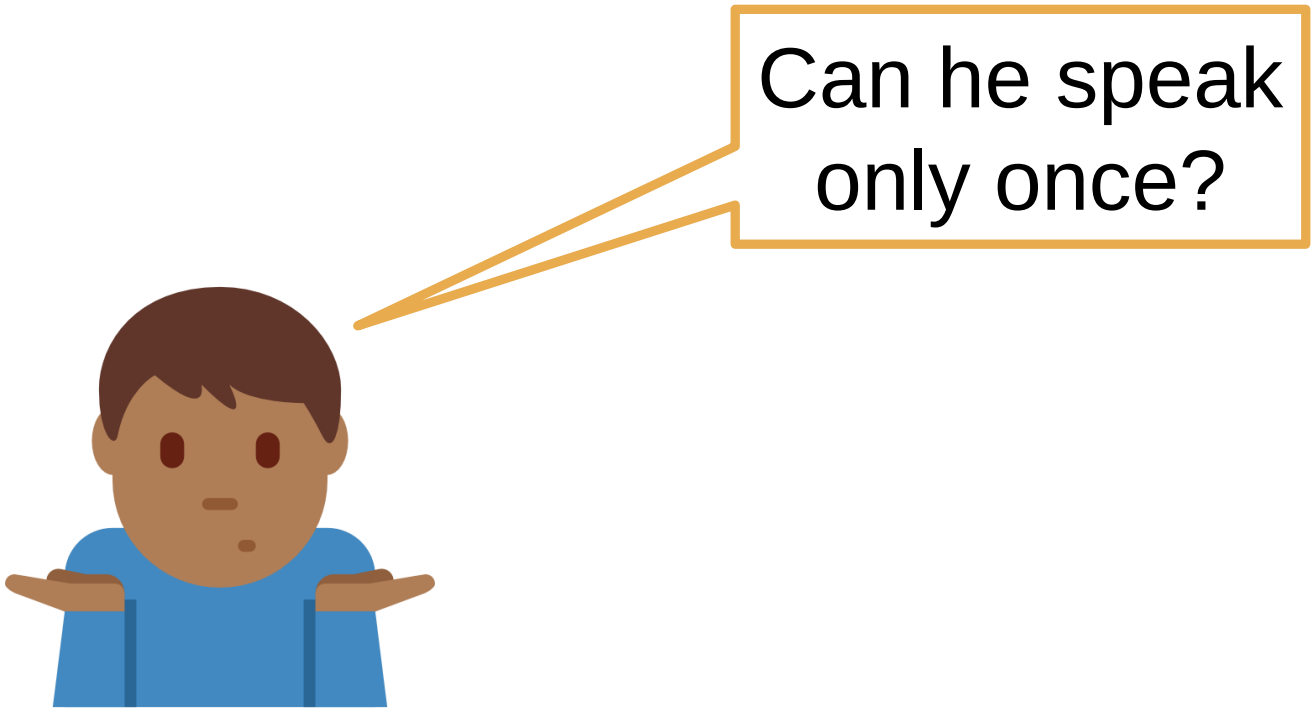
We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable

The true power of printf

We have seen how to make Mr. C

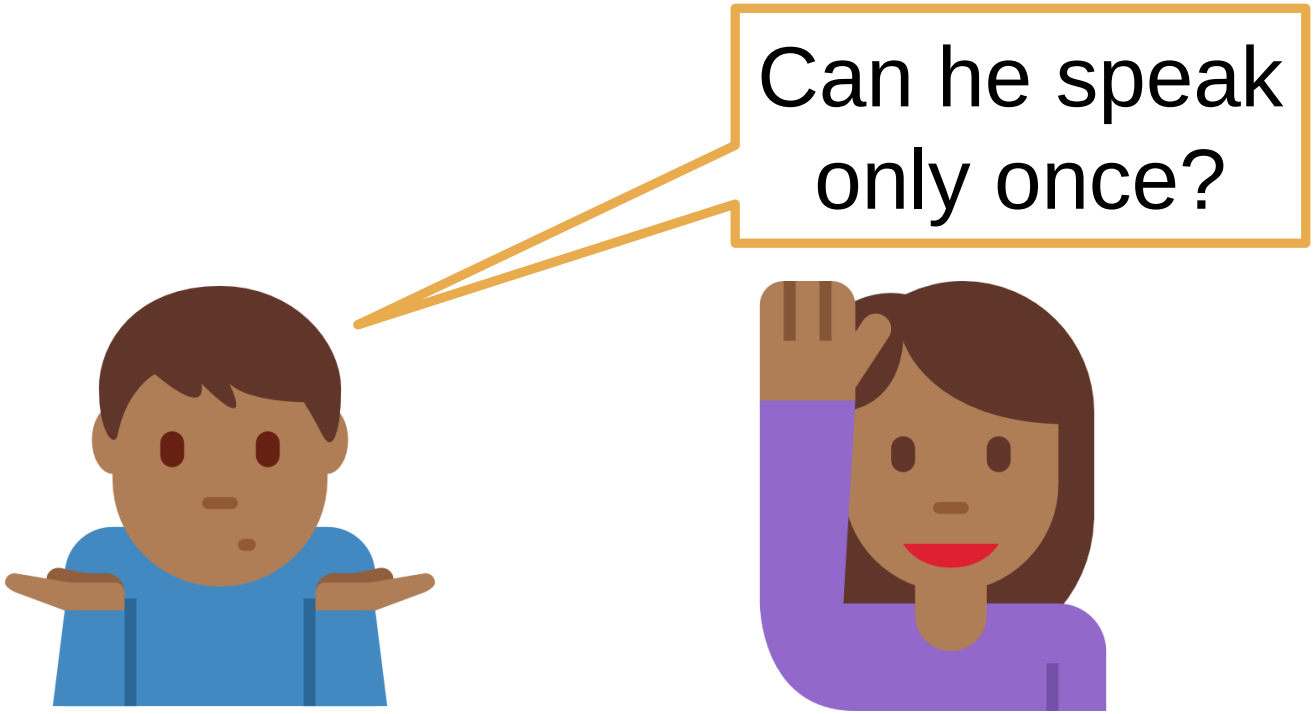
- Say things like “Hello”
- Tell us the value of an integer variable



The true power of printf

We have seen how to make Mr. C

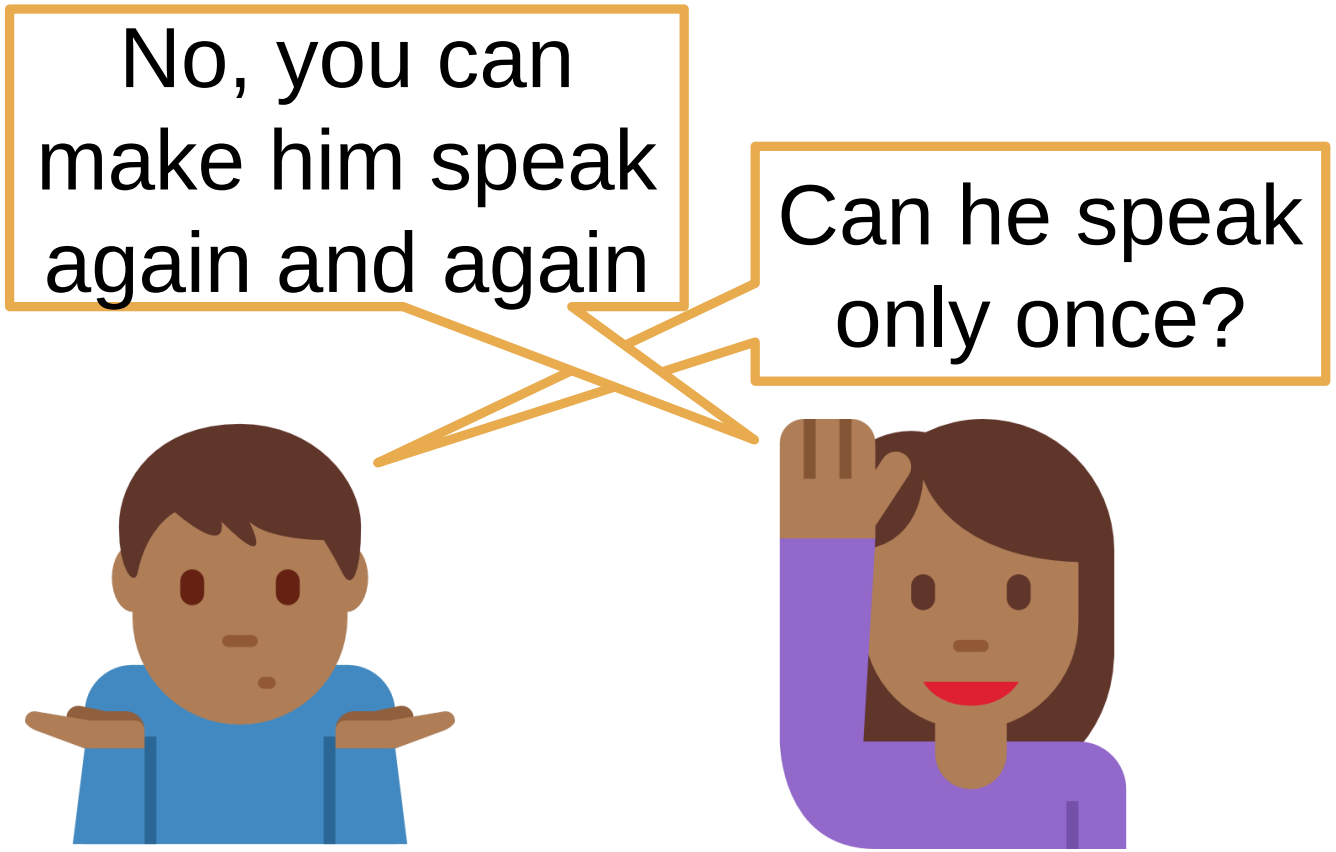
- Say things like “Hello”
- Tell us the value of an integer variable



The true power of printf

We have seen how to make Mr. C

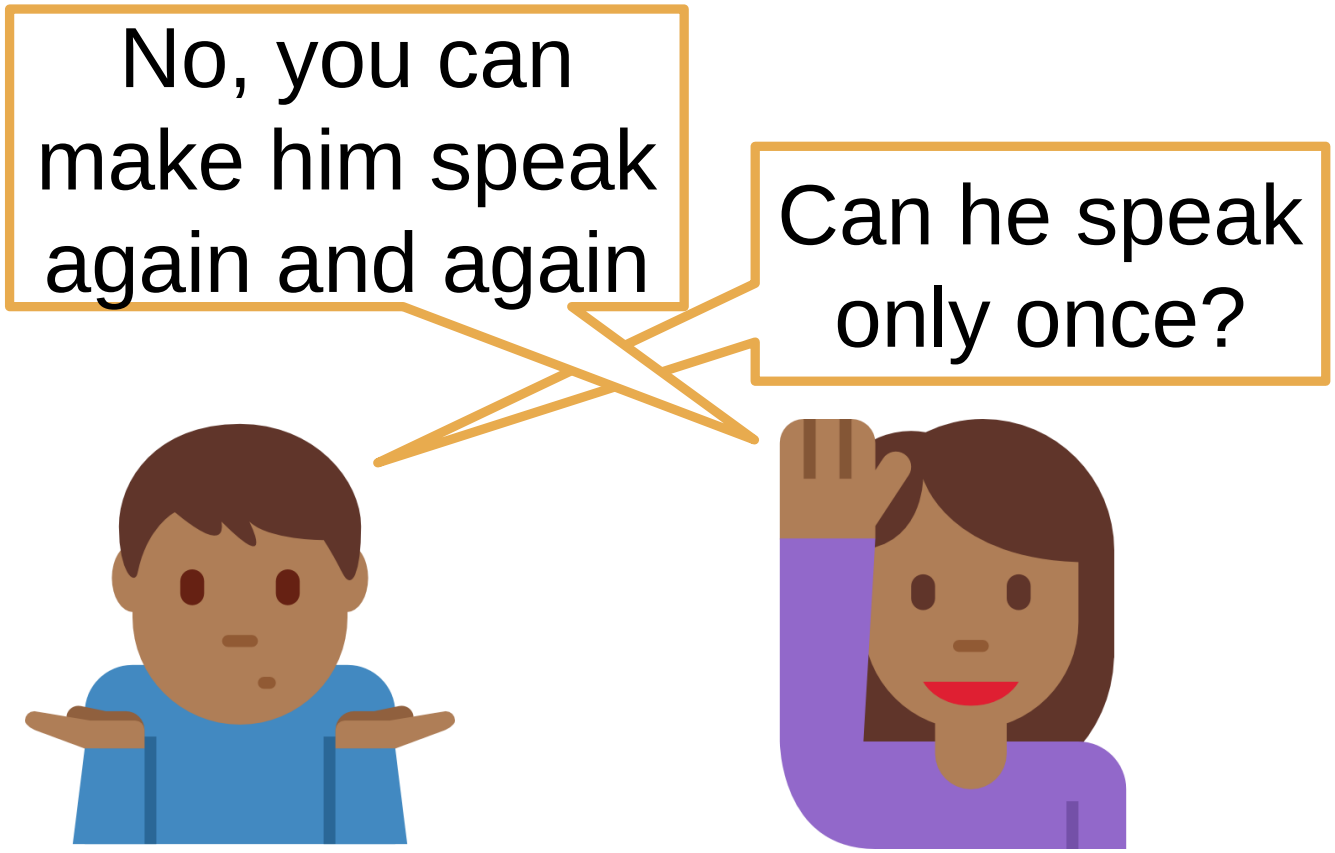
- Say things like “Hello”
- Tell us the value of an integer variable



The true power of printf

We have seen how to make Mr. C

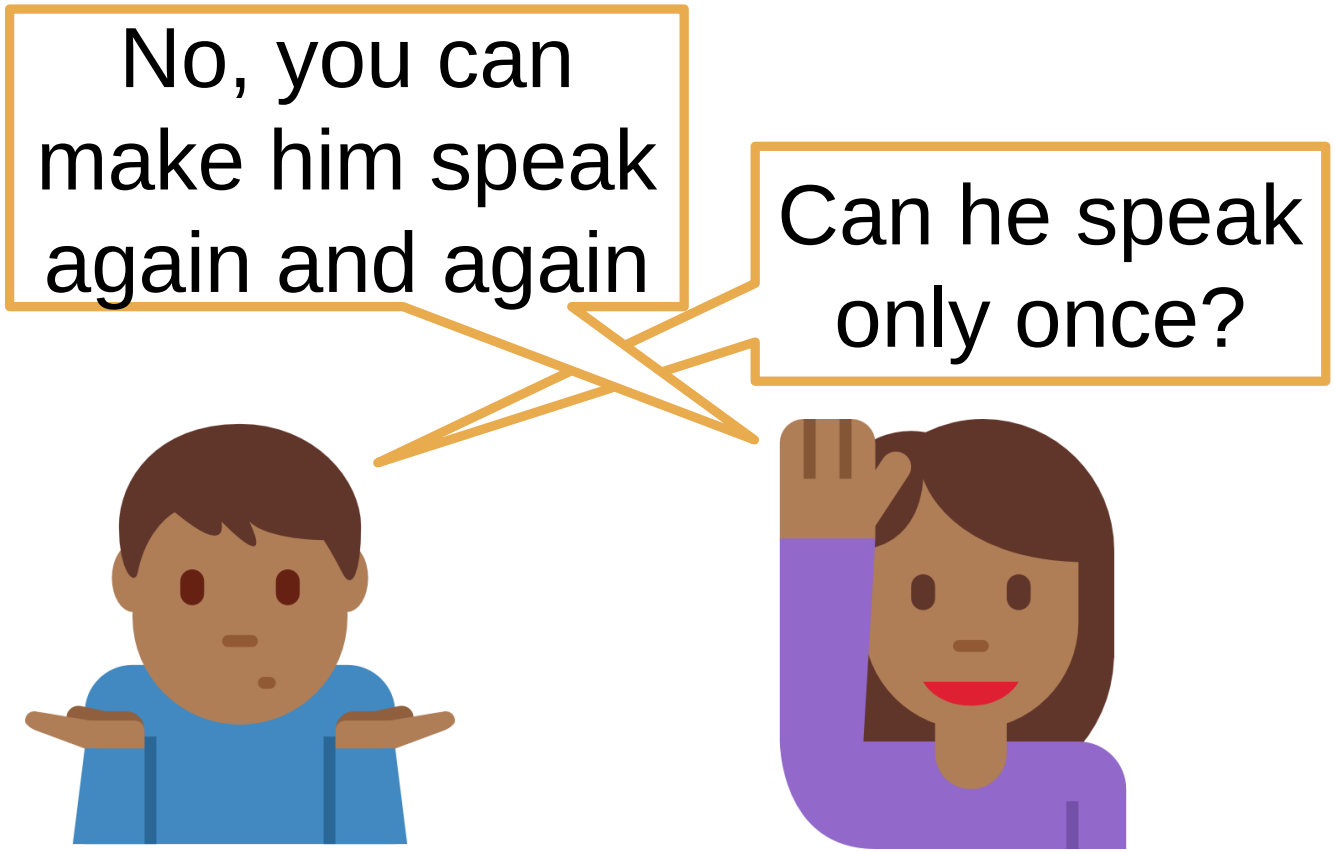
- Say things like “Hello”
- Tell us the value of an integer variable



The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable

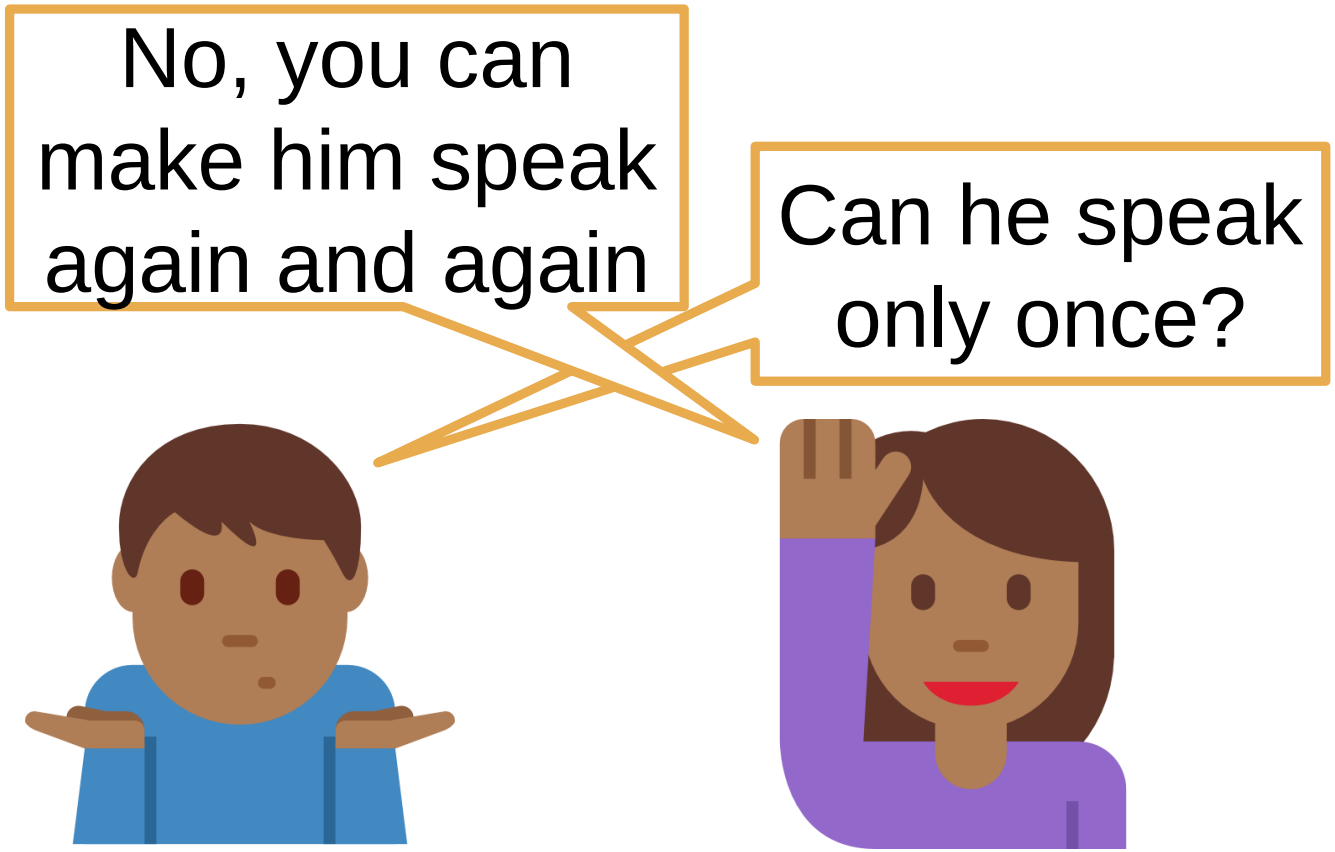



```
#include<stdio.h>
```

The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable

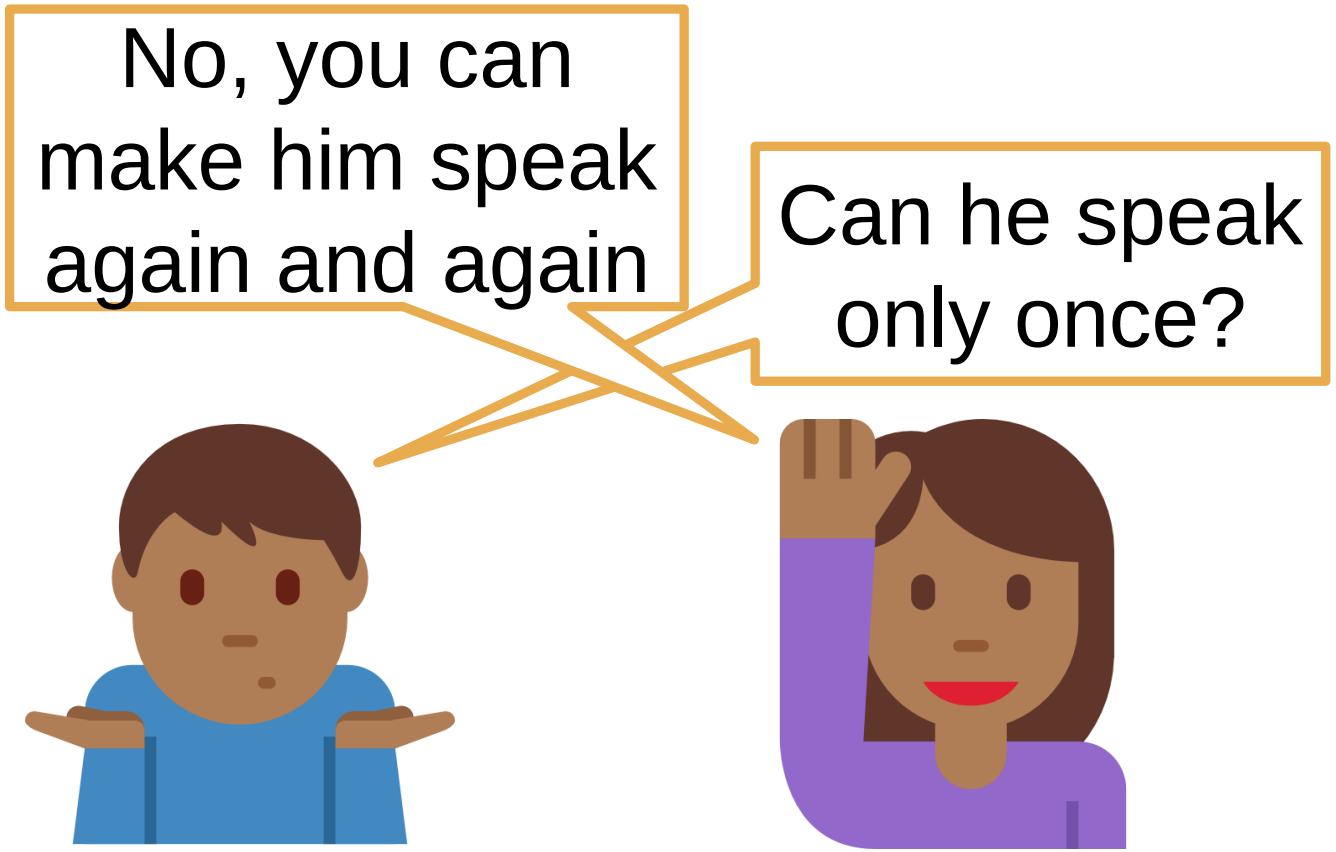



```
#include<stdio.h>
int main(){
```

The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable

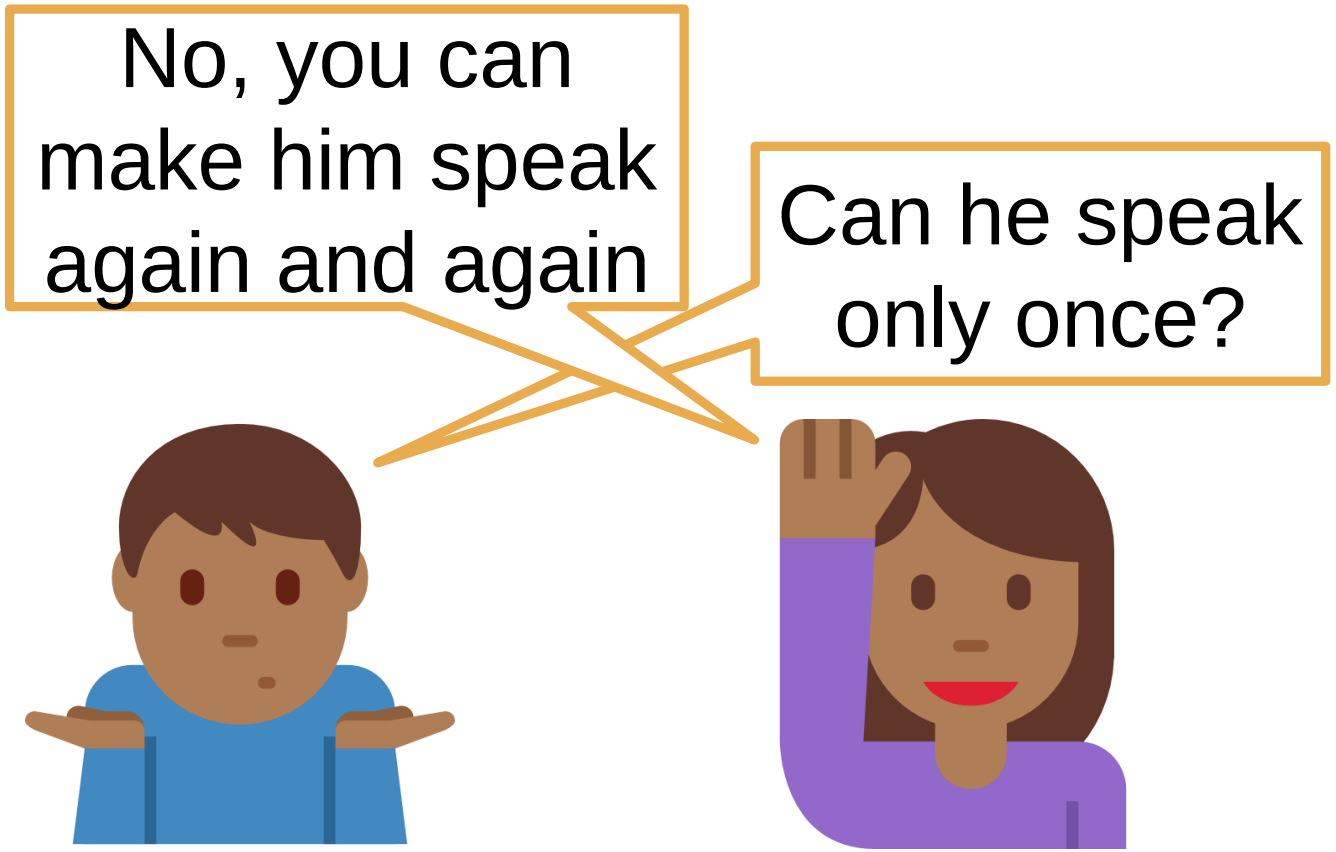



```
#include<stdio.h>
int main(){
int a = 5, b = 4;
```

The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable

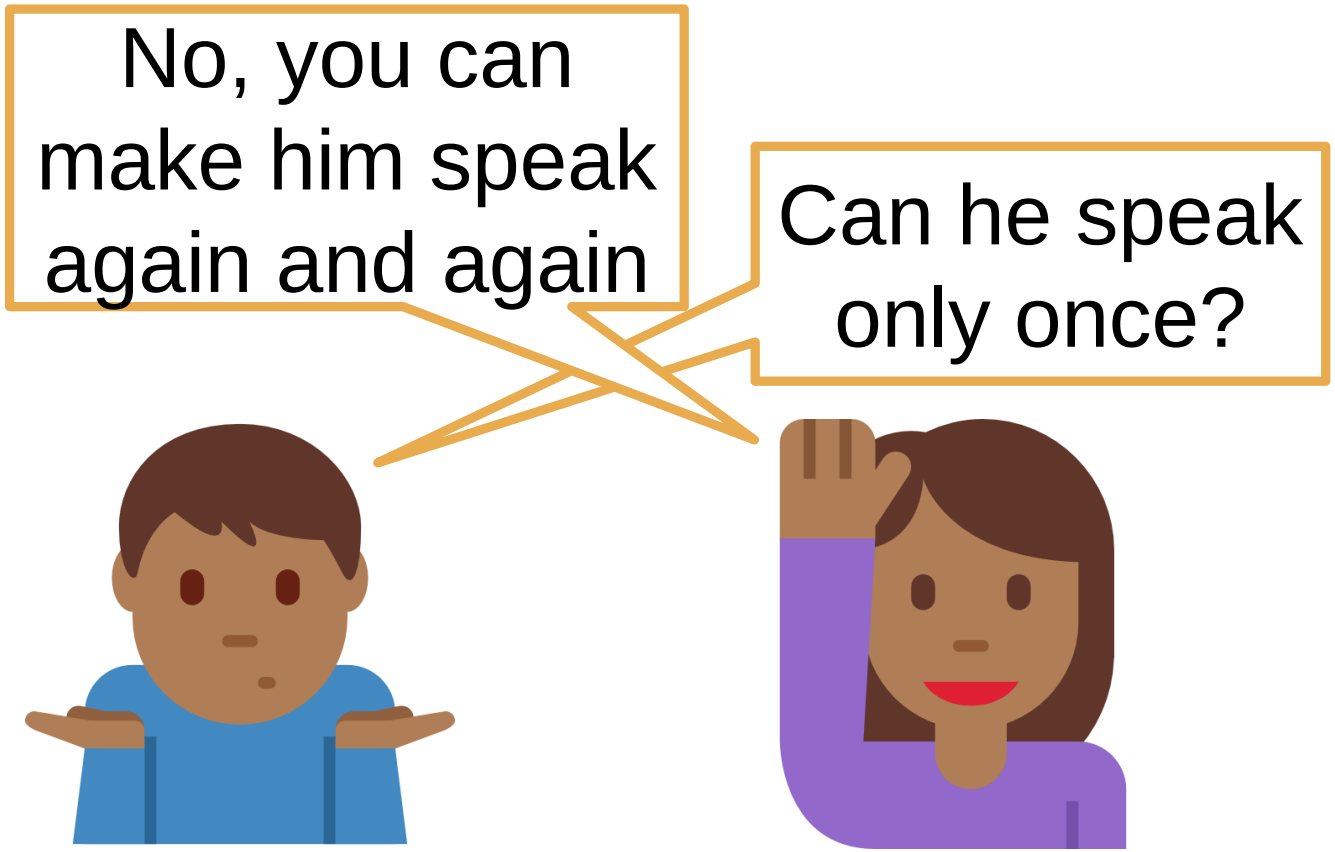



```
#include<stdio.h>
int main(){
int a = 5, b = 4;
printf(“Hello”);
```

The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable

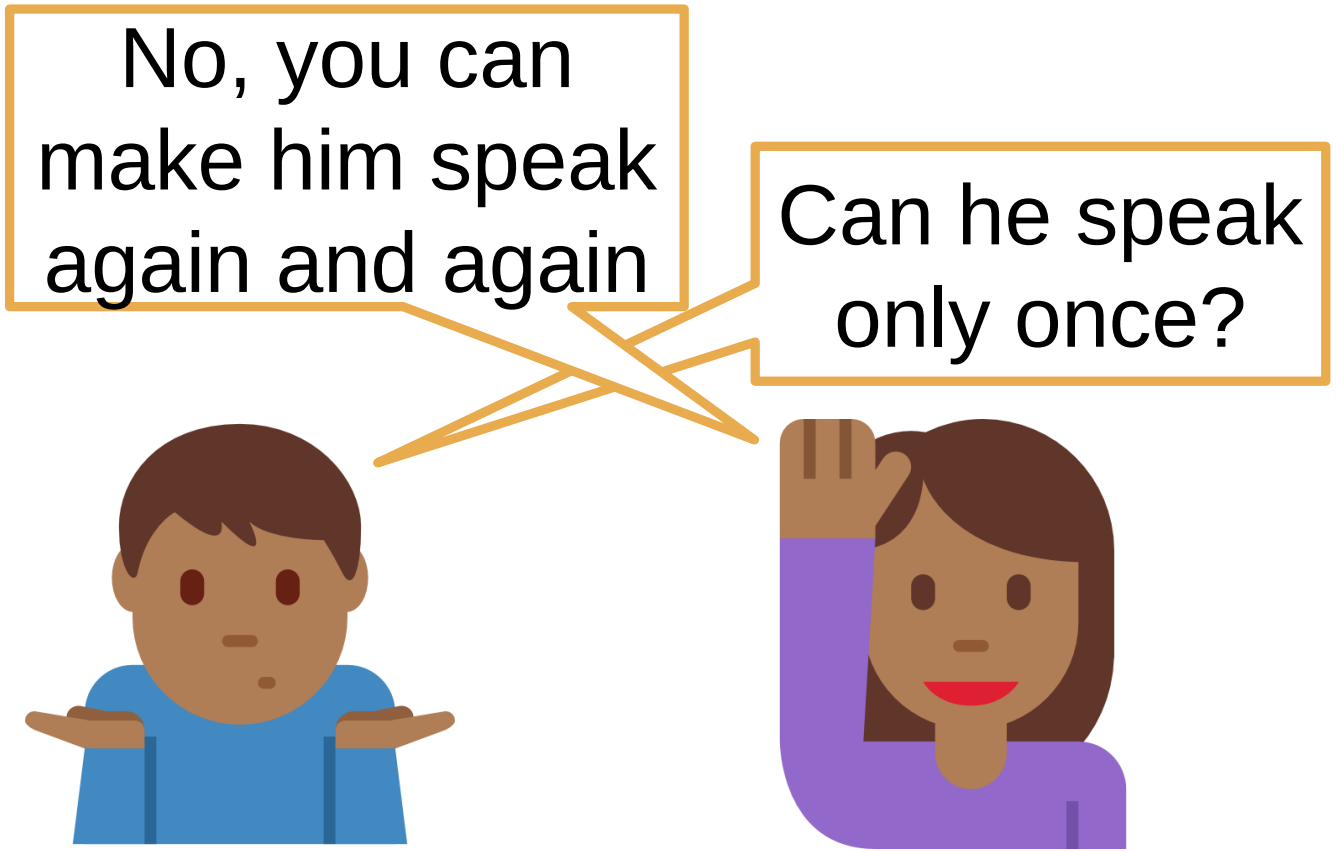



```
#include<stdio.h>
int main(){
int a = 5, b = 4;
printf(“Hello”);
printf(“%d”,a);
```

The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable

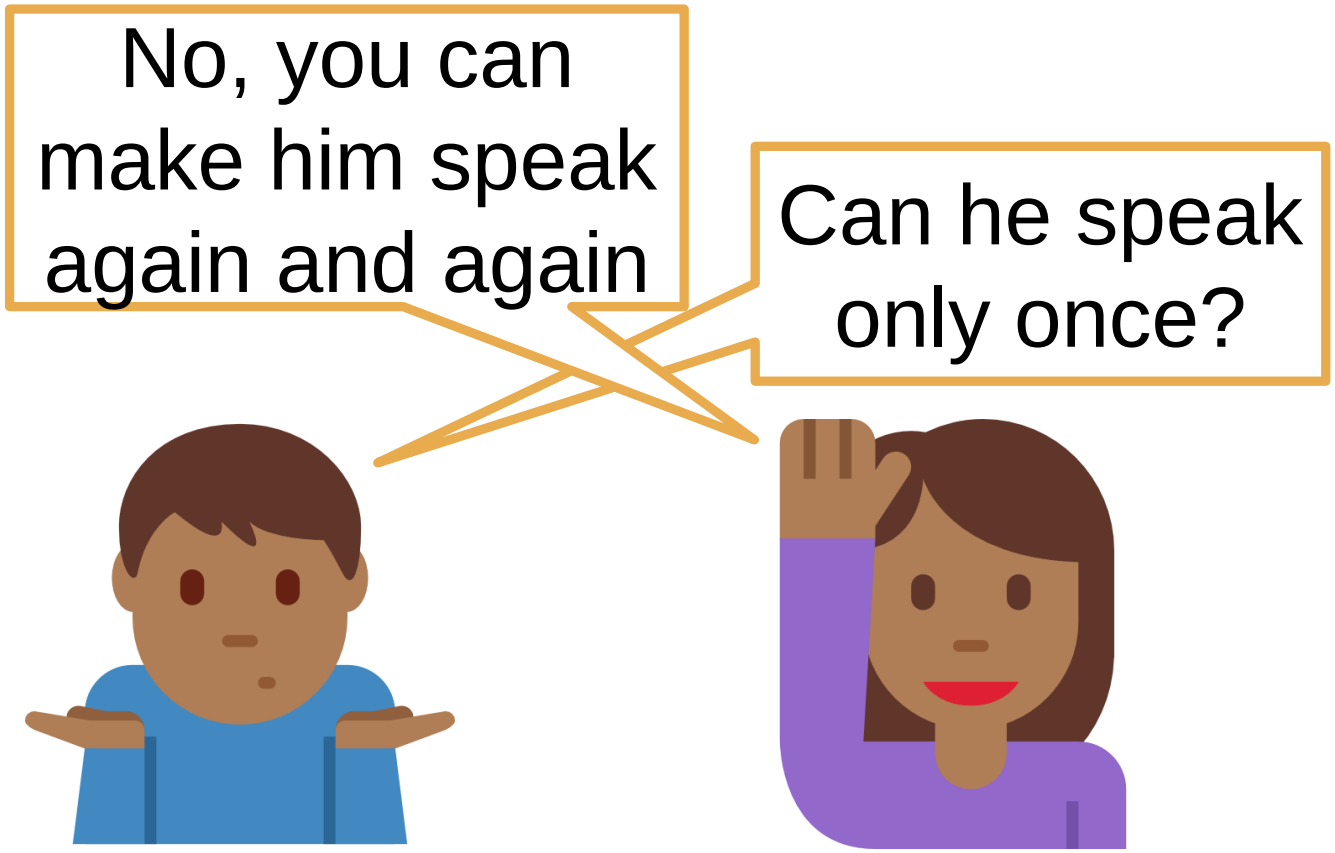



```
#include<stdio.h>
int main(){
int a = 5, b = 4;
printf(“Hello”);
printf(“%d”,a);
printf(“%d”,b);
```

The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable

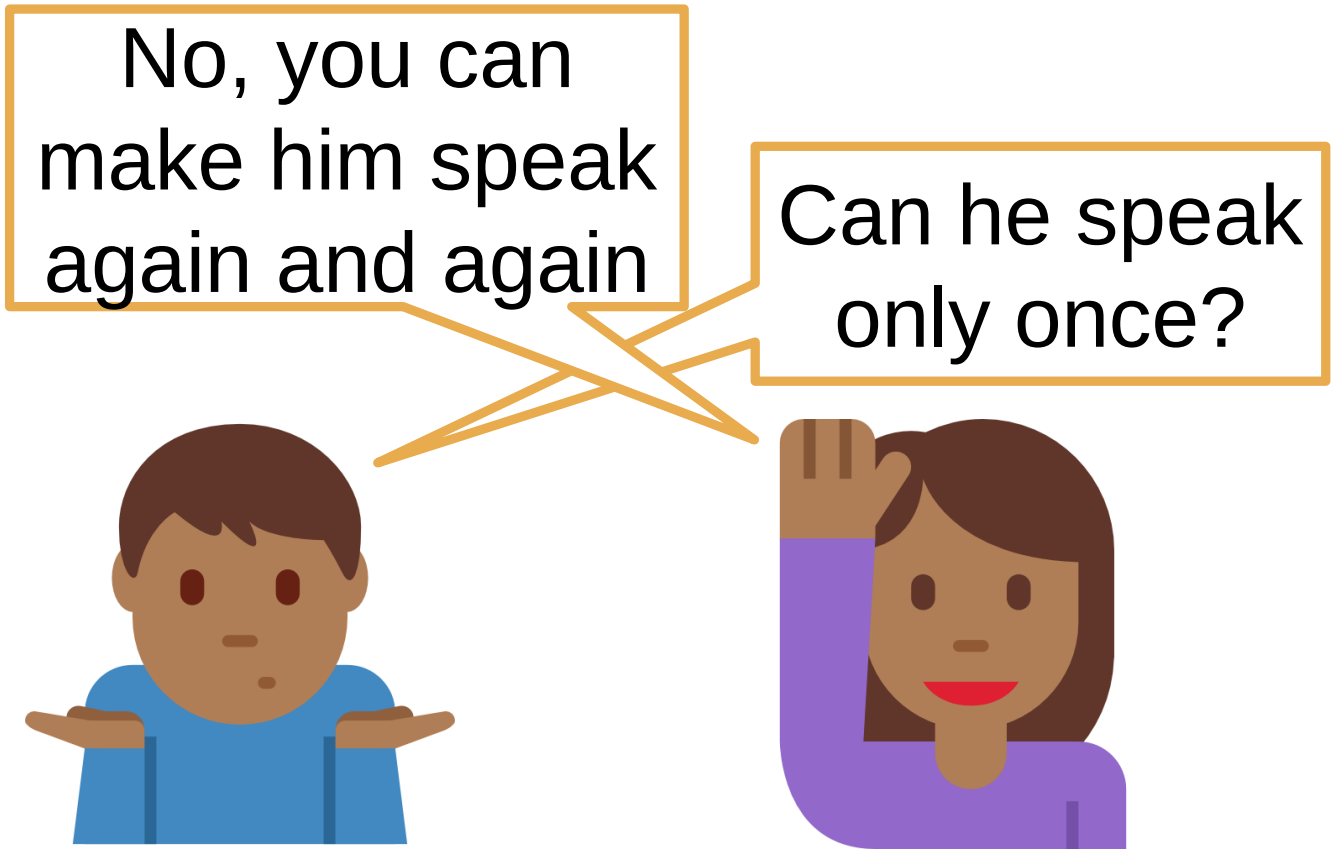



```
#include<stdio.h>
int main(){
int a = 5, b = 4;
printf(“Hello”);
printf(“%d”,a);
printf(“%d”,b);
return 0;
```

The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable

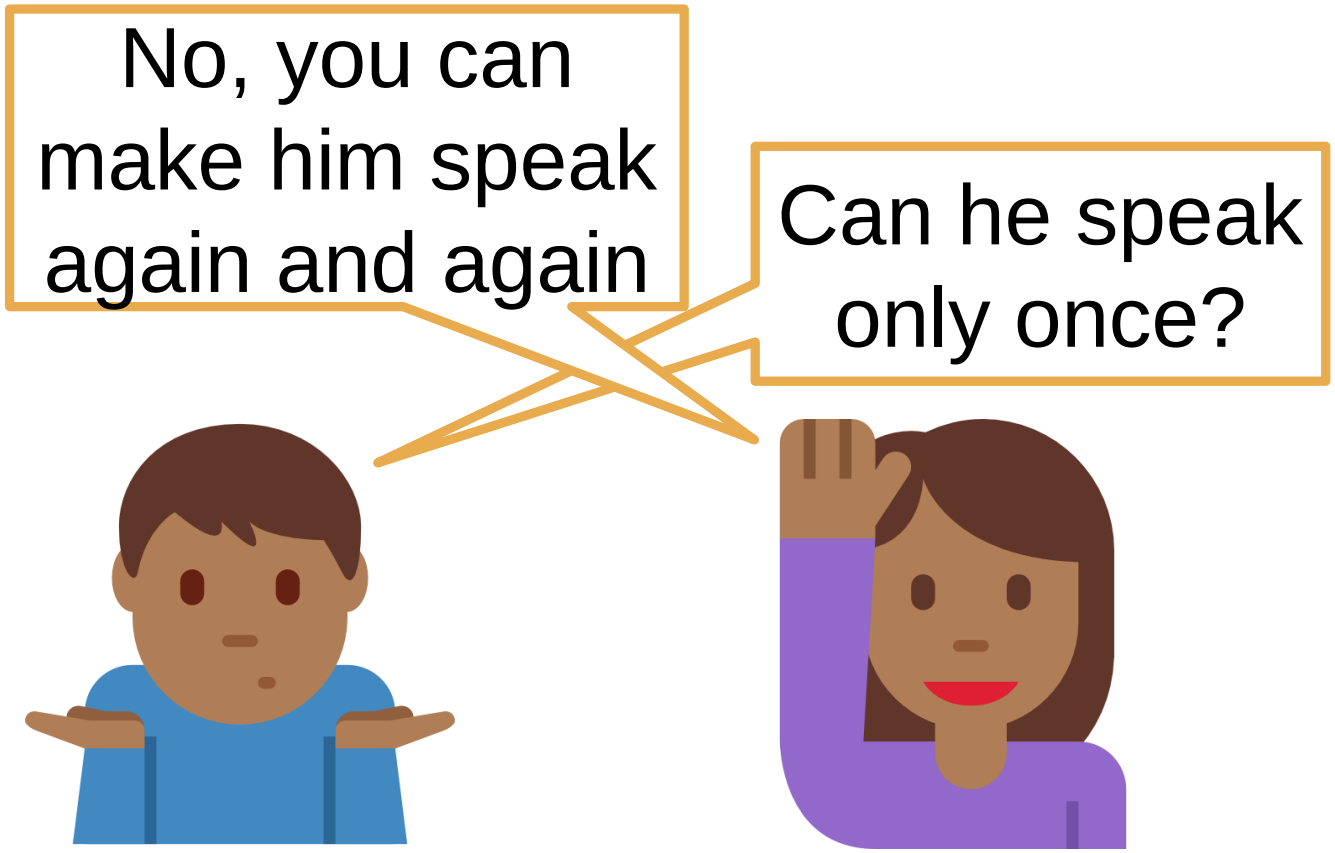



```
#include<stdio.h>
int main(){
int a = 5, b = 4;
printf(“Hello”);
printf(“%d”,a);
printf(“%d”,b);
return 0;
}
```

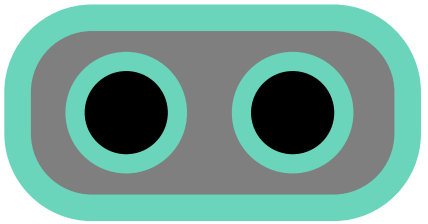
The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable



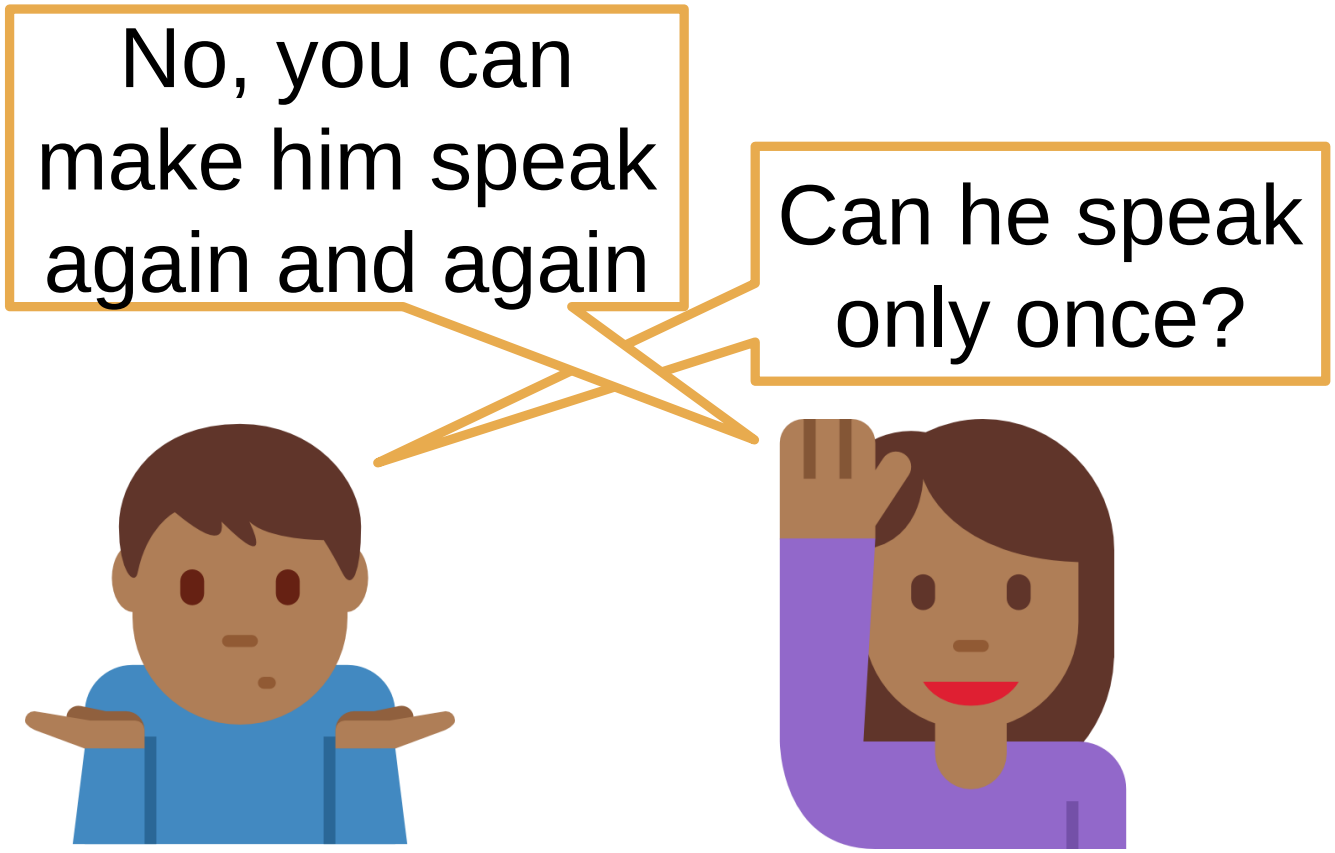

```
#include<stdio.h>
int main(){
int a = 5, b = 4;
printf(“Hello”);
printf(“%d”,a);
printf(“%d”,b);
return 0;
}
```



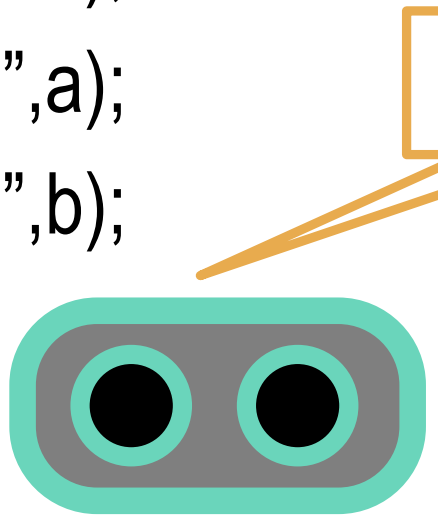
The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable



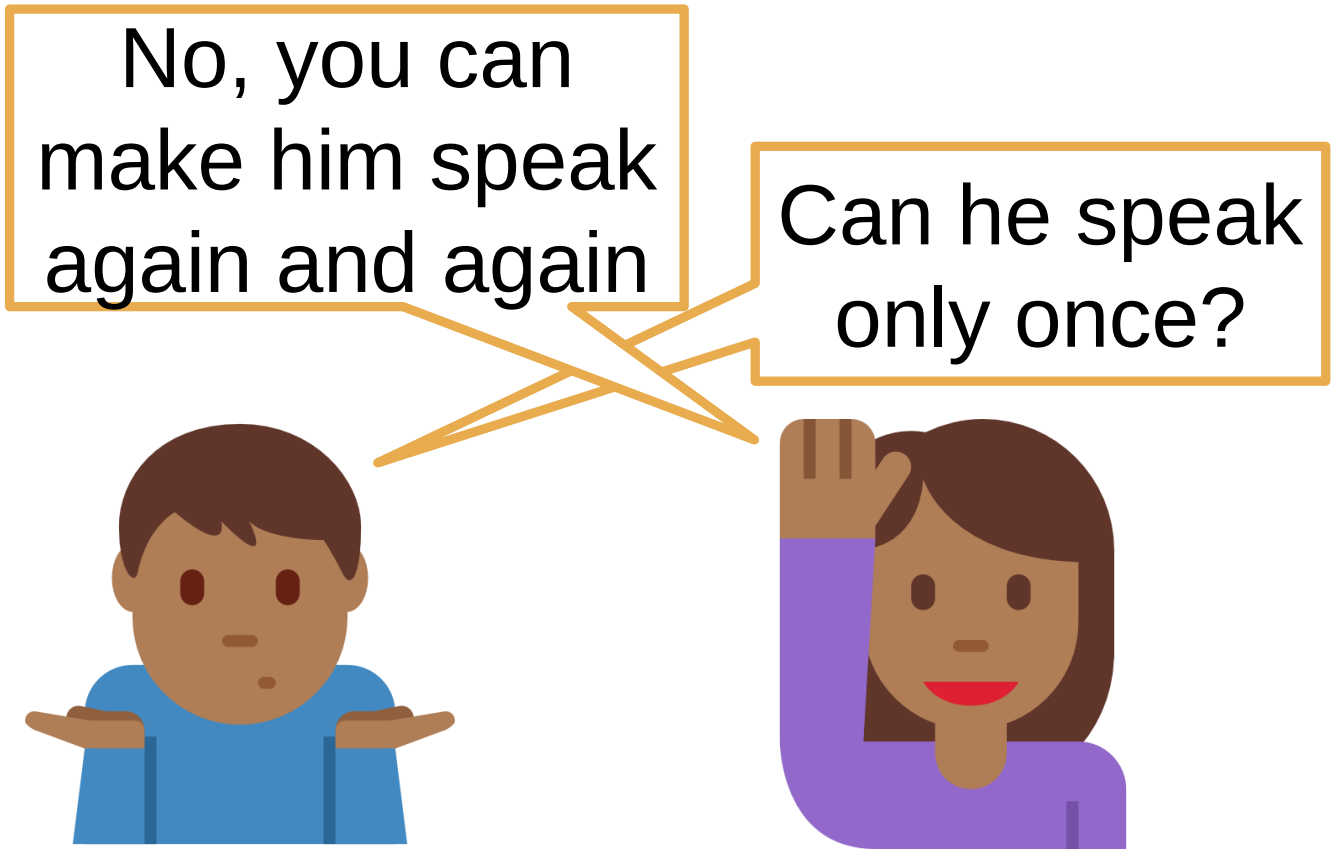

```
#include<stdio.h>
int main(){
int a = 5, b = 4;
printf(“Hello”);
printf(“%d”,a);
printf(“%d”,b);
return 0;
}
```



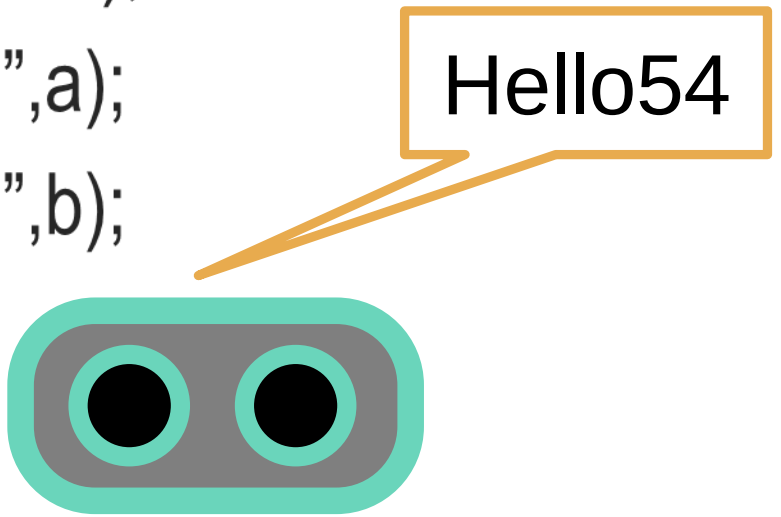
The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable



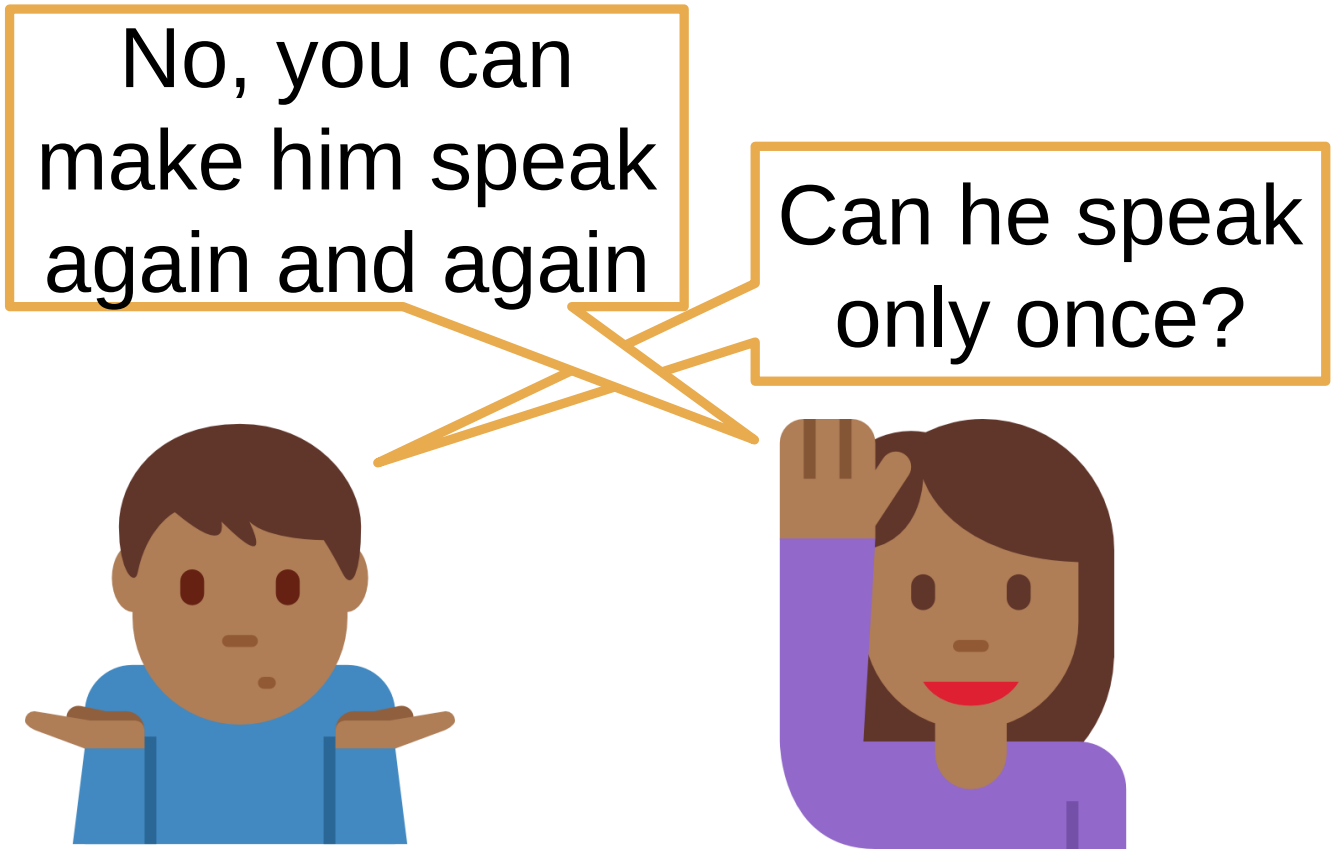

```
#include<stdio.h>
int main(){
int a = 5, b = 4;
printf(“Hello ”);
printf(“%d”,a);
printf(“%d”,b);
return 0;
}
```



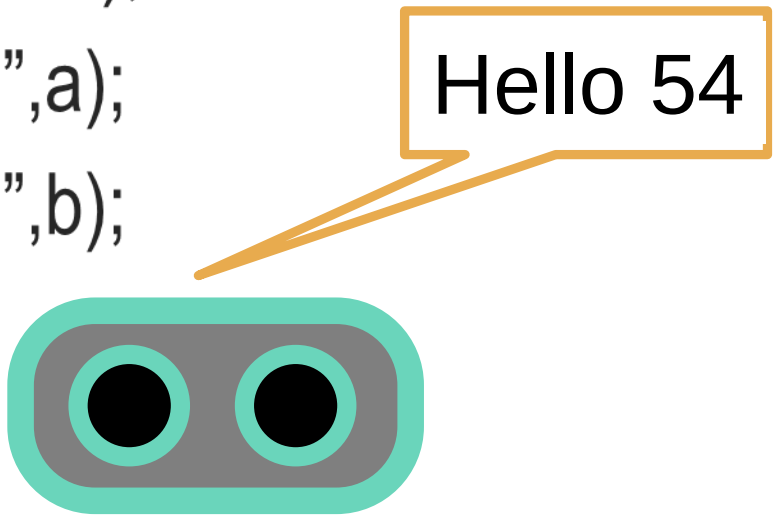
The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable




```
#include<stdio.h>
int main(){
int a = 5, b = 4;
printf(“Hello ”);
printf(“%d”,a);
printf(“%d”,b);
return 0;
}
```



The true power of printf

We have seen how to make Mr. C

Say things like "Hell

Tell us the value of

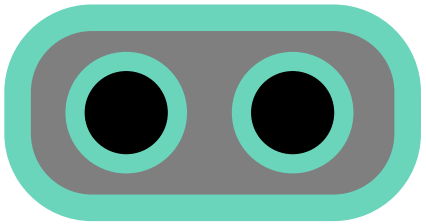
Don't be afraid to experiment

No, you can make him speak again and again

Can he speak only once?

```
#include<stdio.h>
int main(){
int a = 5, b = 4;
printf("Hello ");
printf("%d",a);
printf("%d",b);
return 0;
}
```

Hello 54



The true power of printf

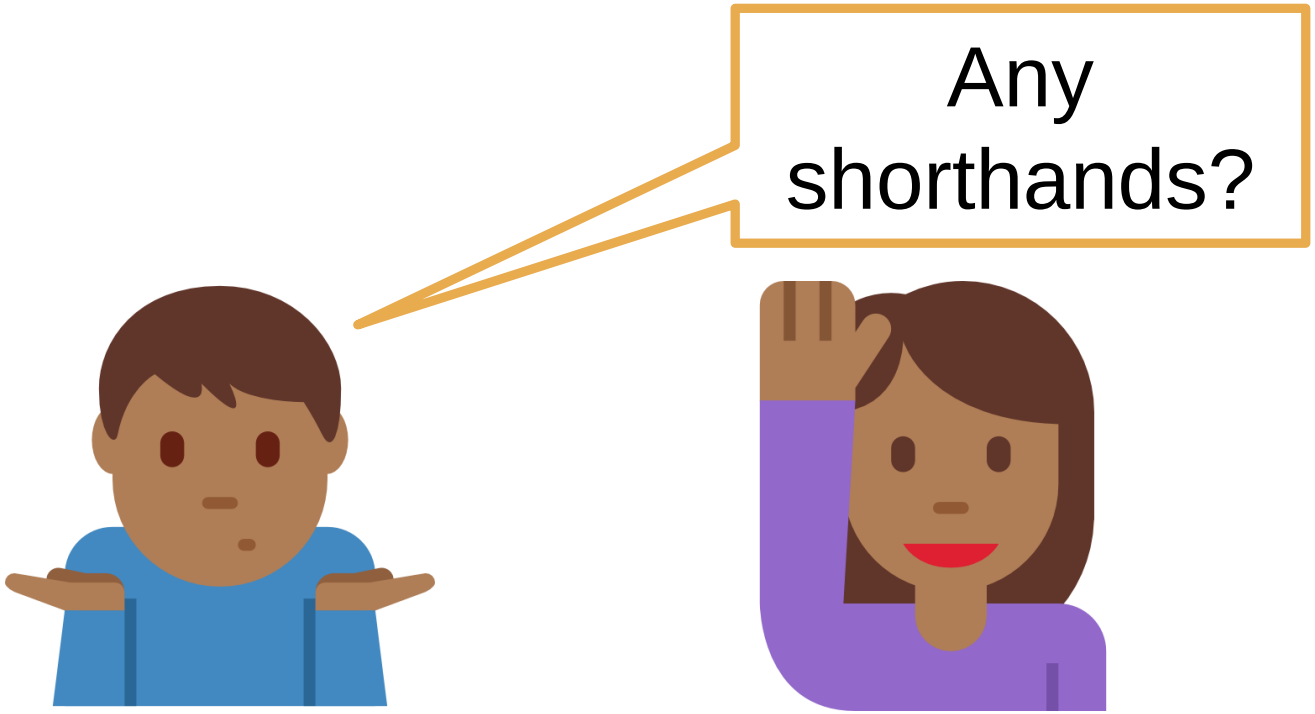
We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable

The true power of printf

We have seen how to make Mr. C

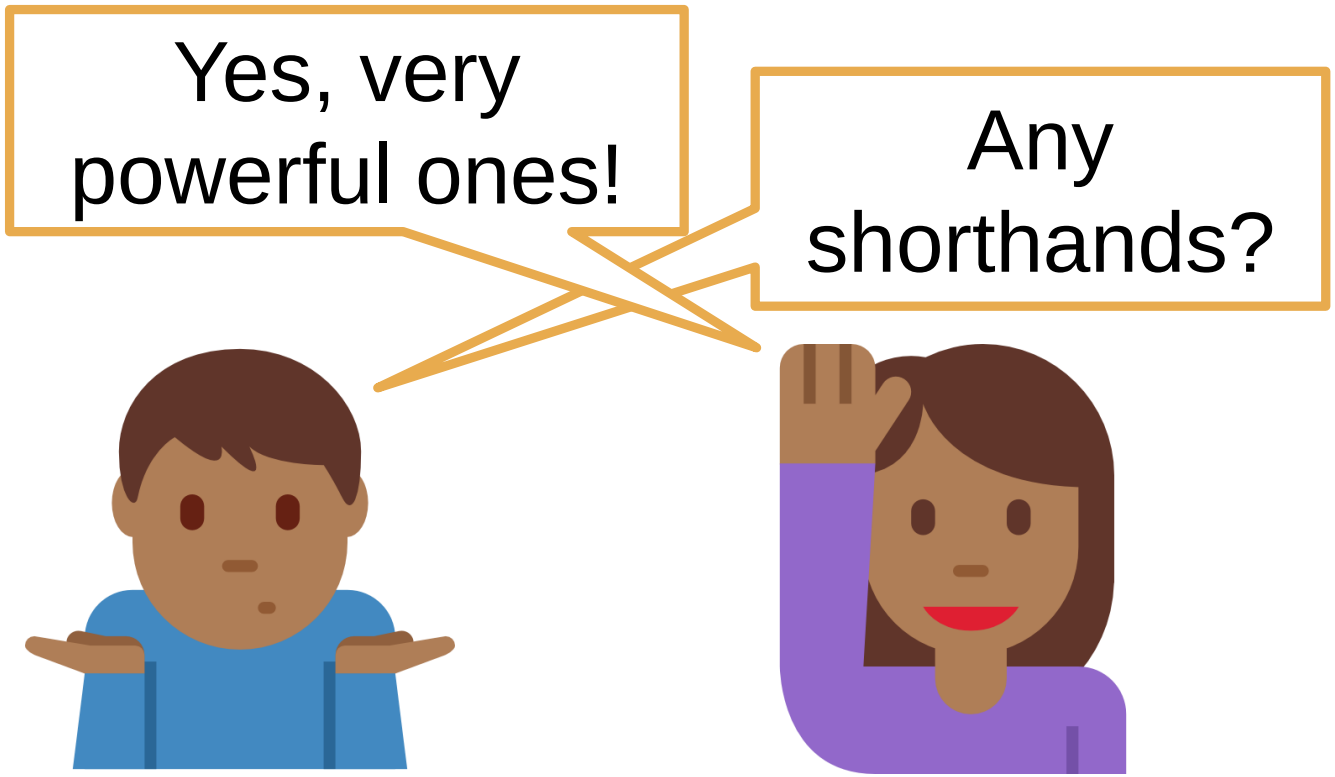
- Say things like “Hello”
- Tell us the value of an integer variable



The true power of printf

We have seen how to make Mr. C

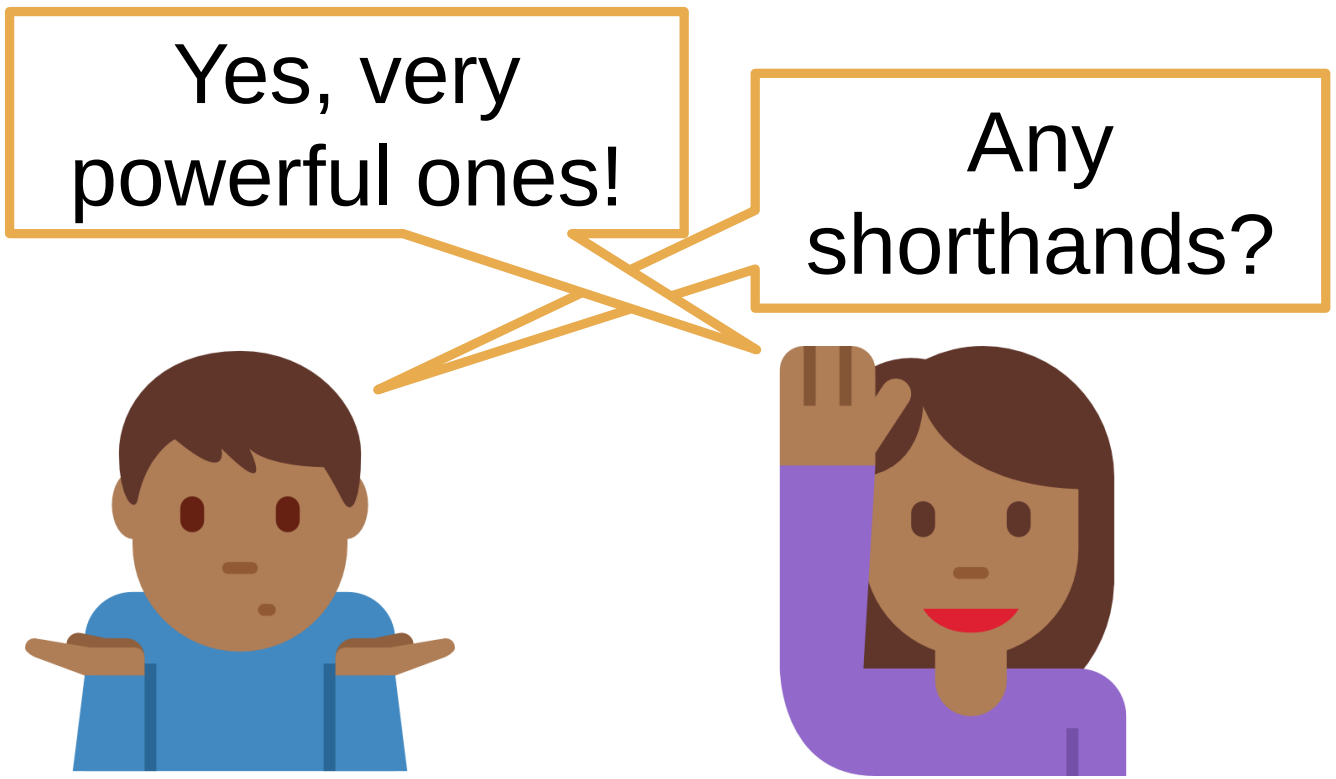
- Say things like “Hello”
- Tell us the value of an integer variable



The true power of printf

We have seen how to make Mr. C

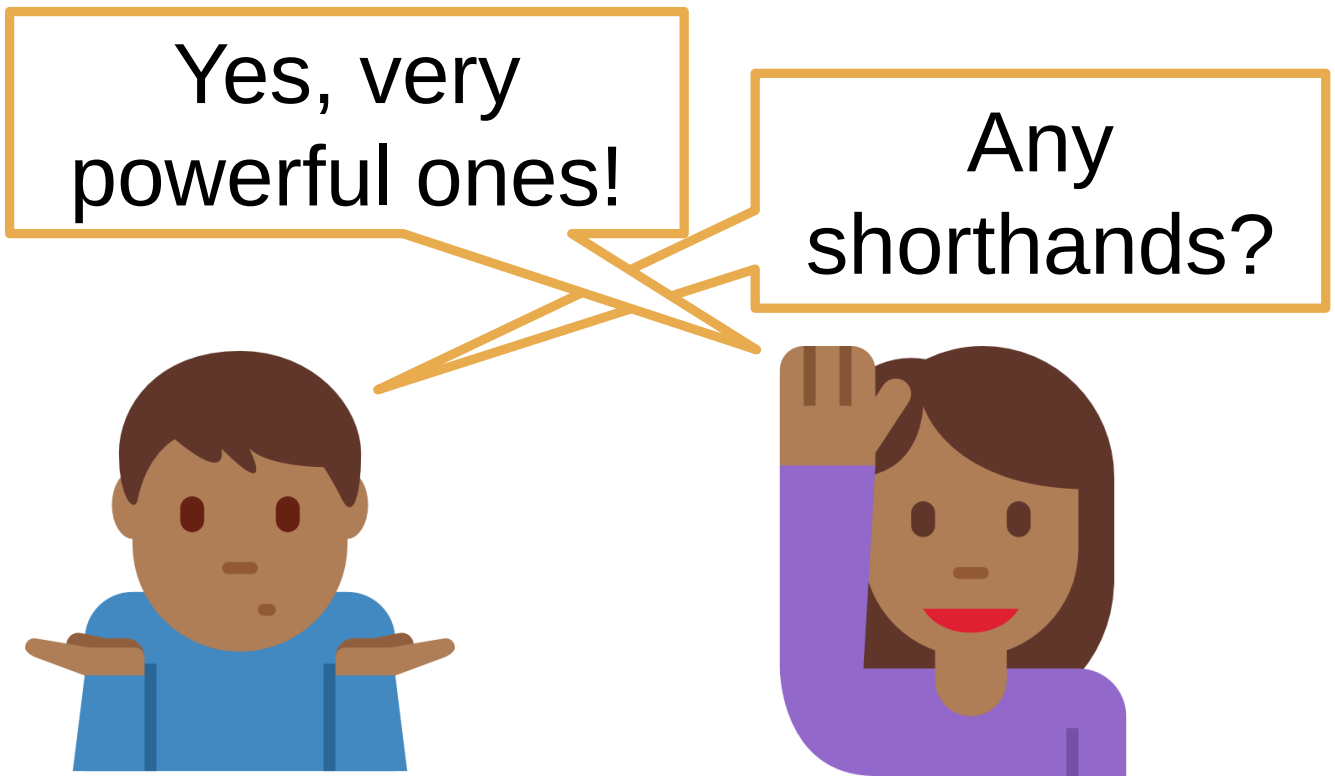
- Say things like “Hello”
- Tell us the value of an integer variable



The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable

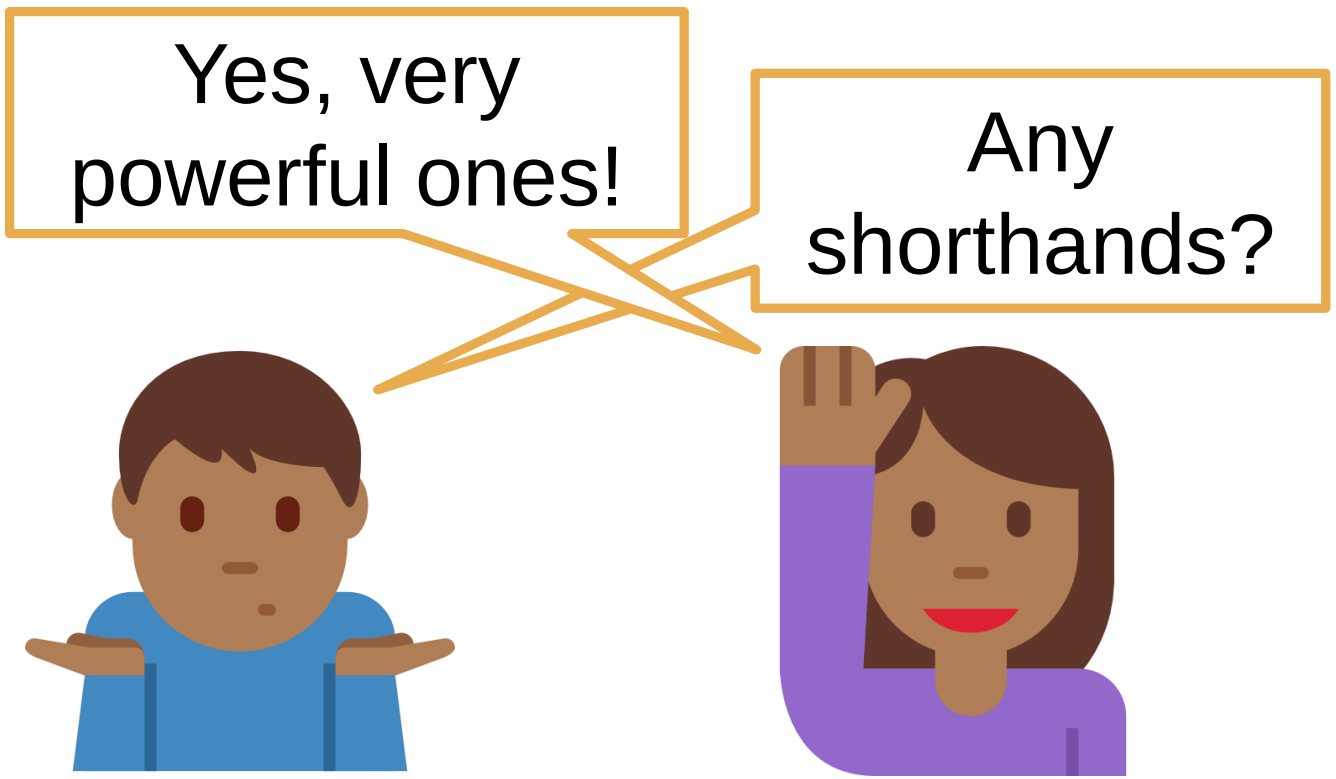



```
#include<stdio.h>
```

The true power of printf

We have seen how to make Mr. C

- Say things like "Hello"
- Tell us the value of an integer variable

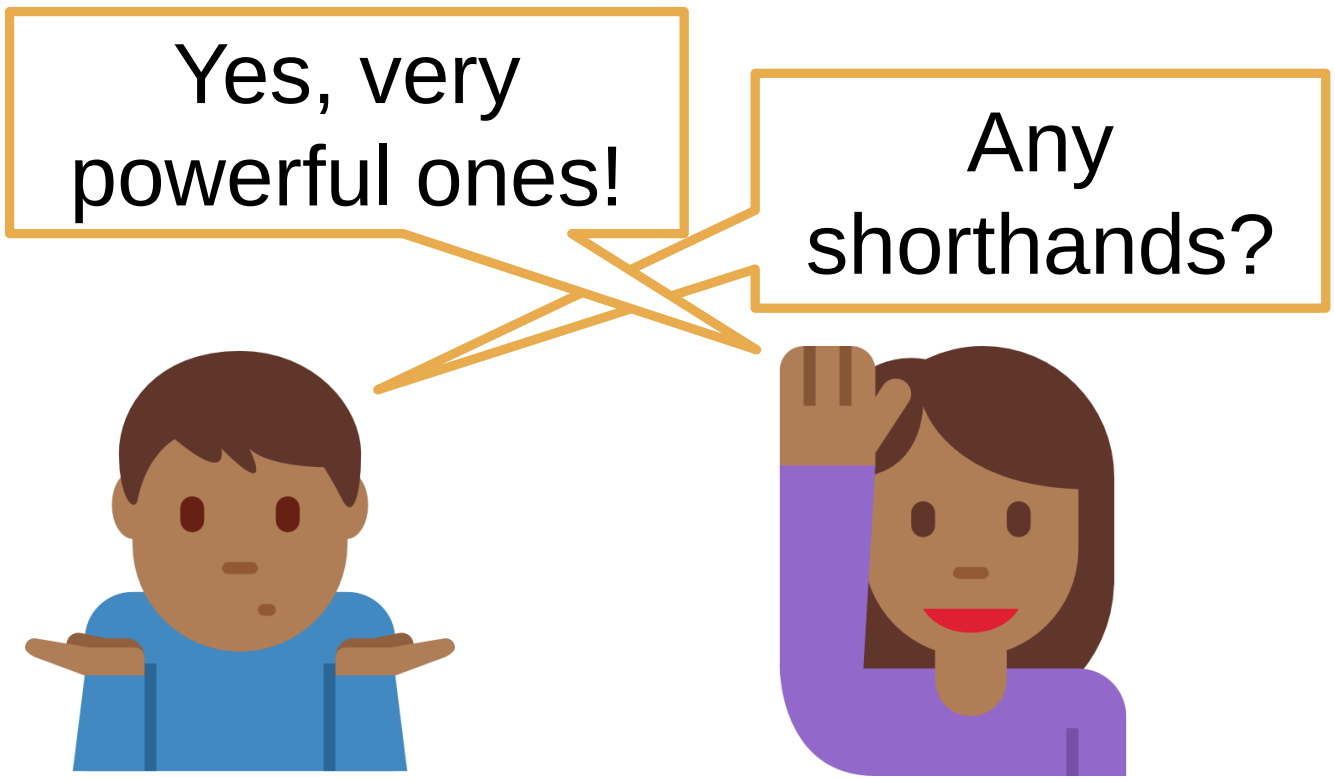



```
#include<stdio.h>
int main(){
```

The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable

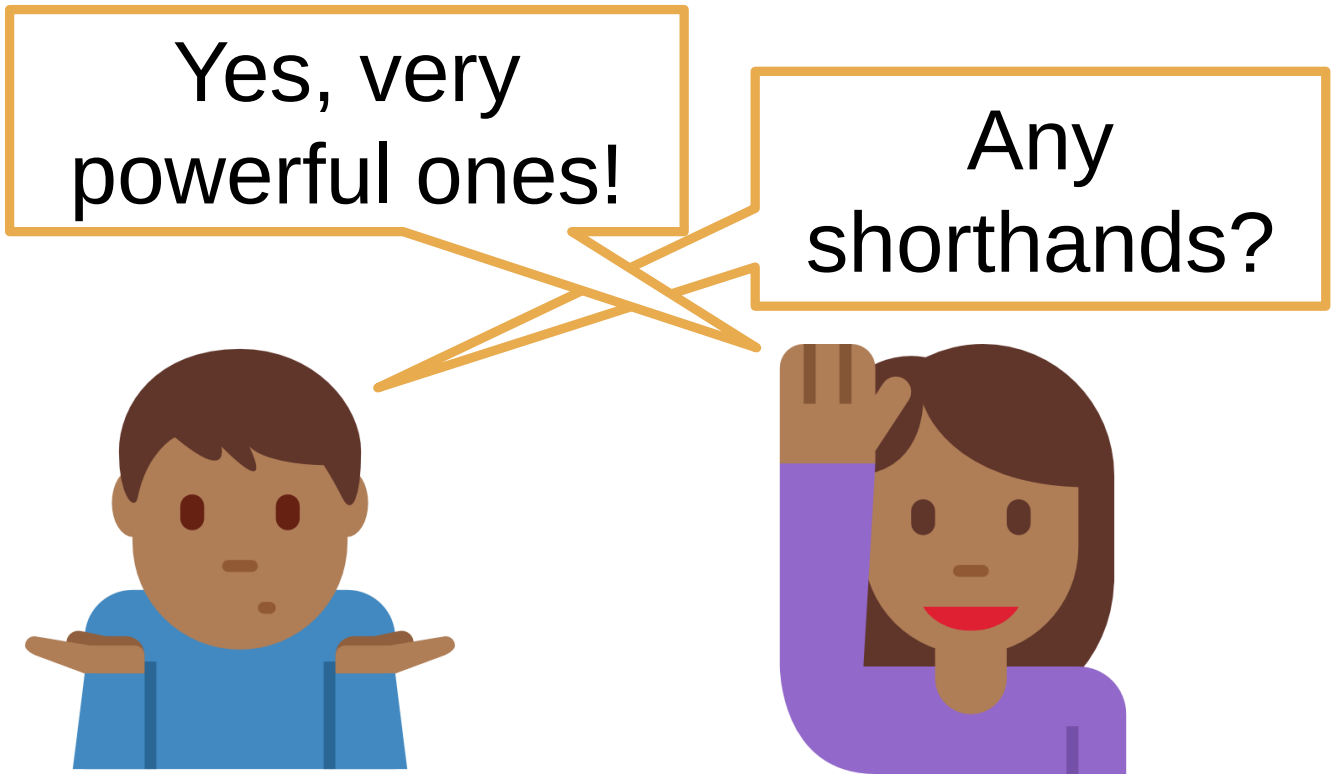



```
#include<stdio.h>
int main(){
int a = 5, b = 4;
```

The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable

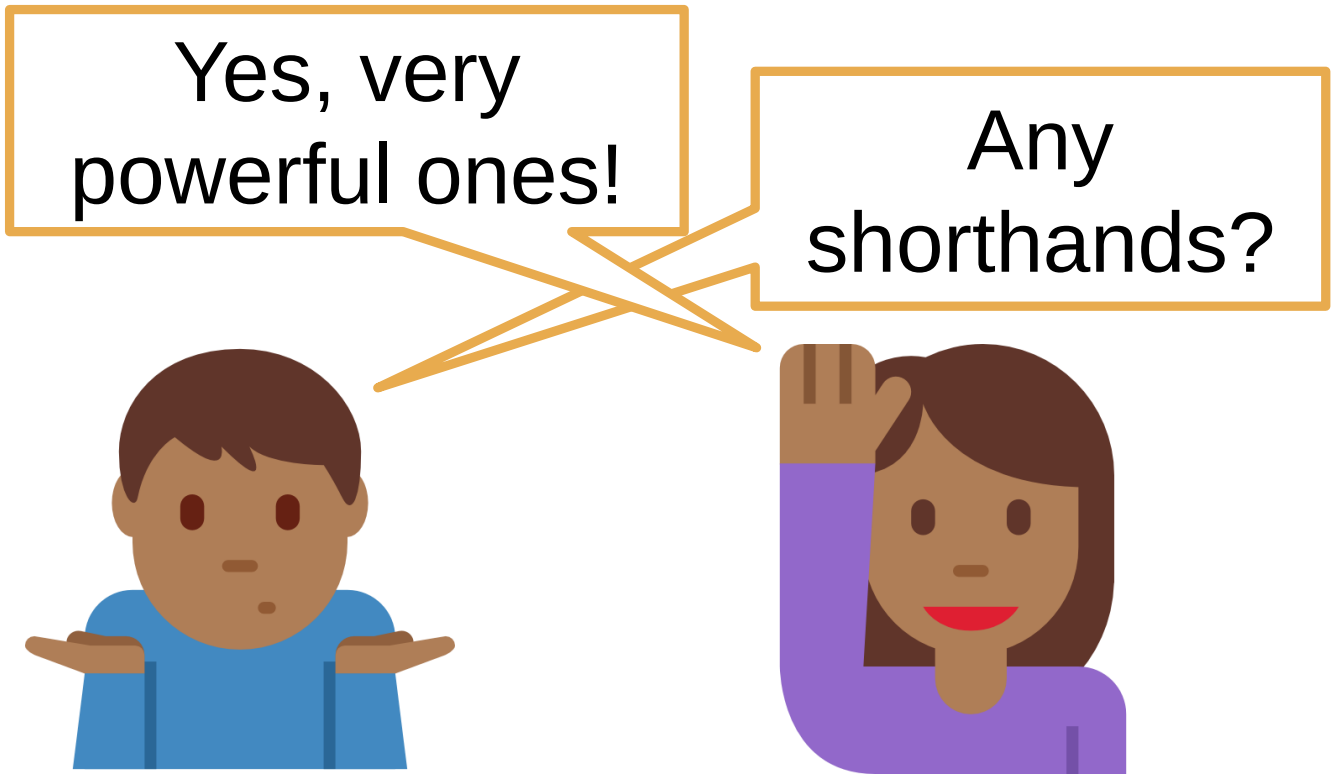



```
#include<stdio.h>
int main(){
int a = 5, b = 4;
printf(“Hello %d %d”,a,b);
```

The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable

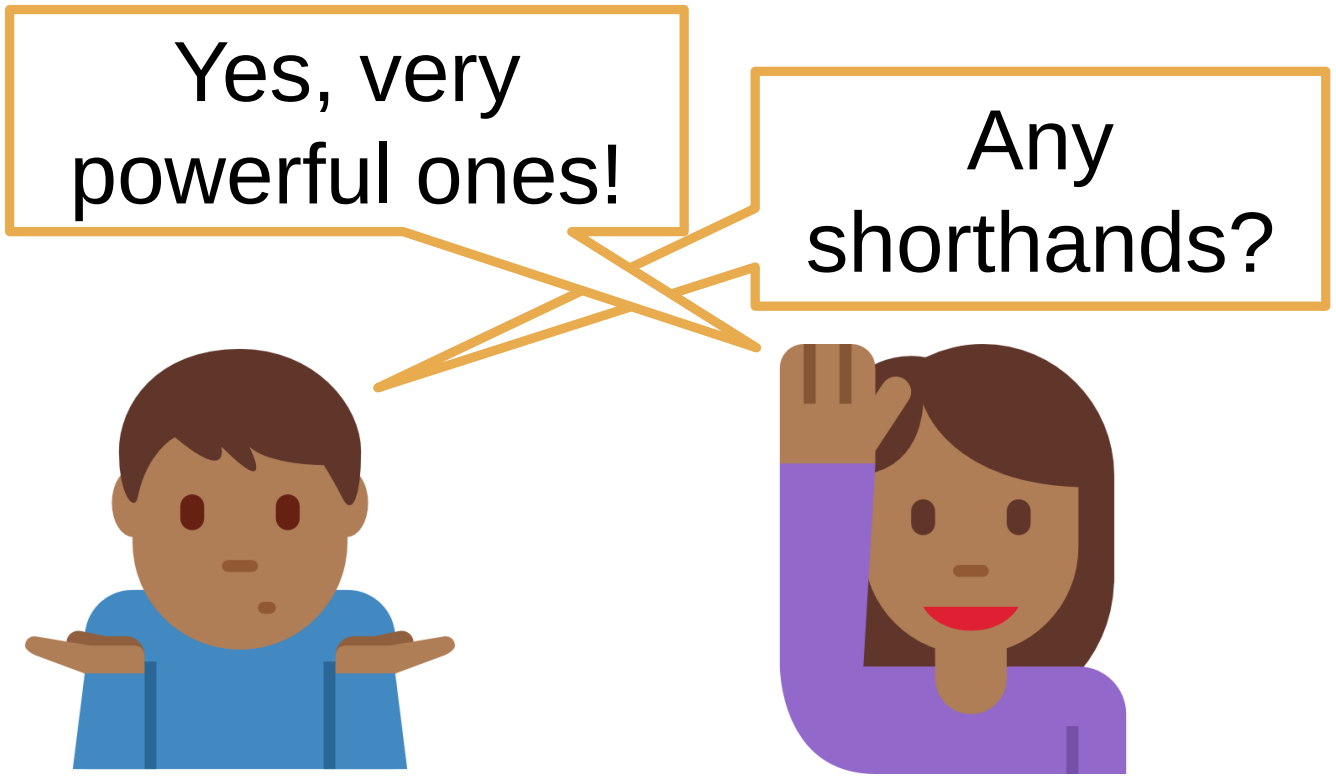



```
#include<stdio.h>
int main(){
int a = 5, b = 4;
printf(“Hello %d %d”,a,b);
return 0;
```

The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable

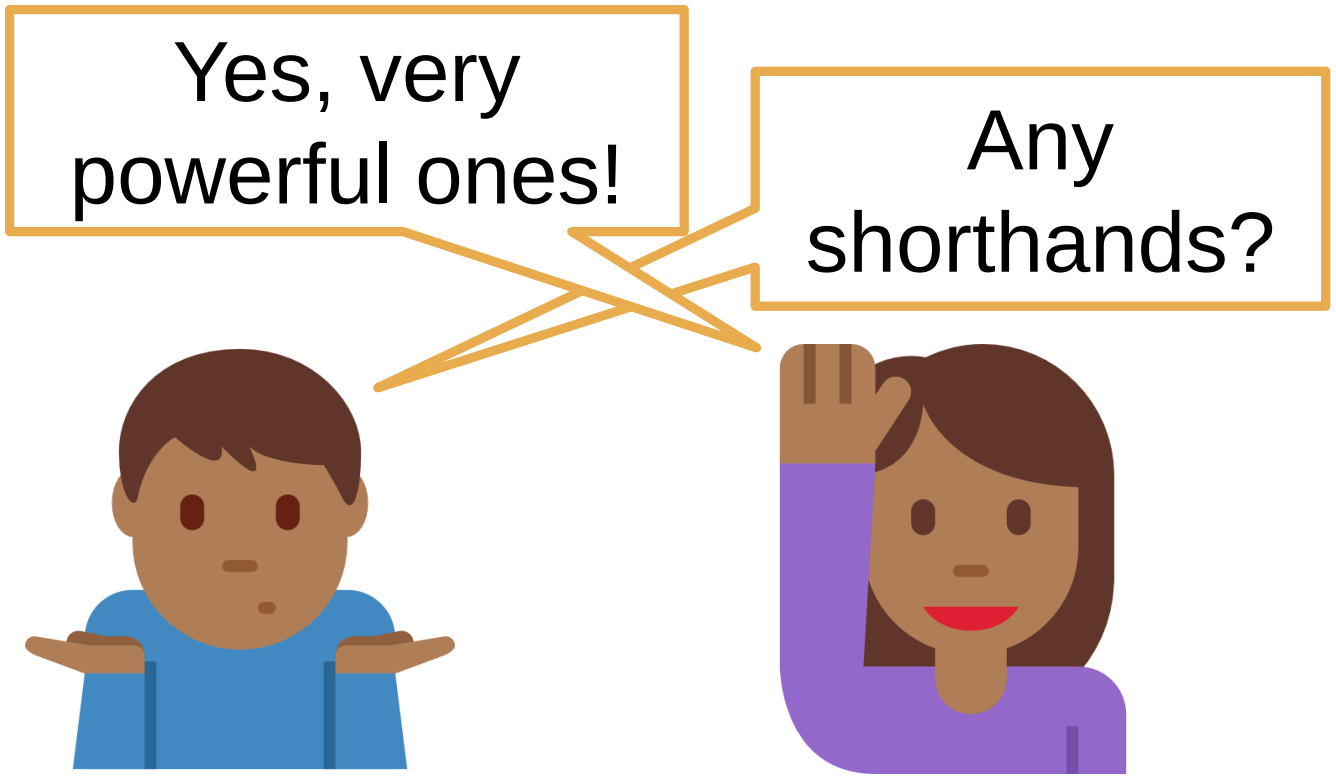



```
#include<stdio.h>
int main(){
int a = 5, b = 4;
printf(“Hello %d %d”,a,b);
return 0;
}
```

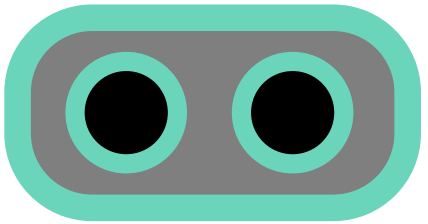
The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable



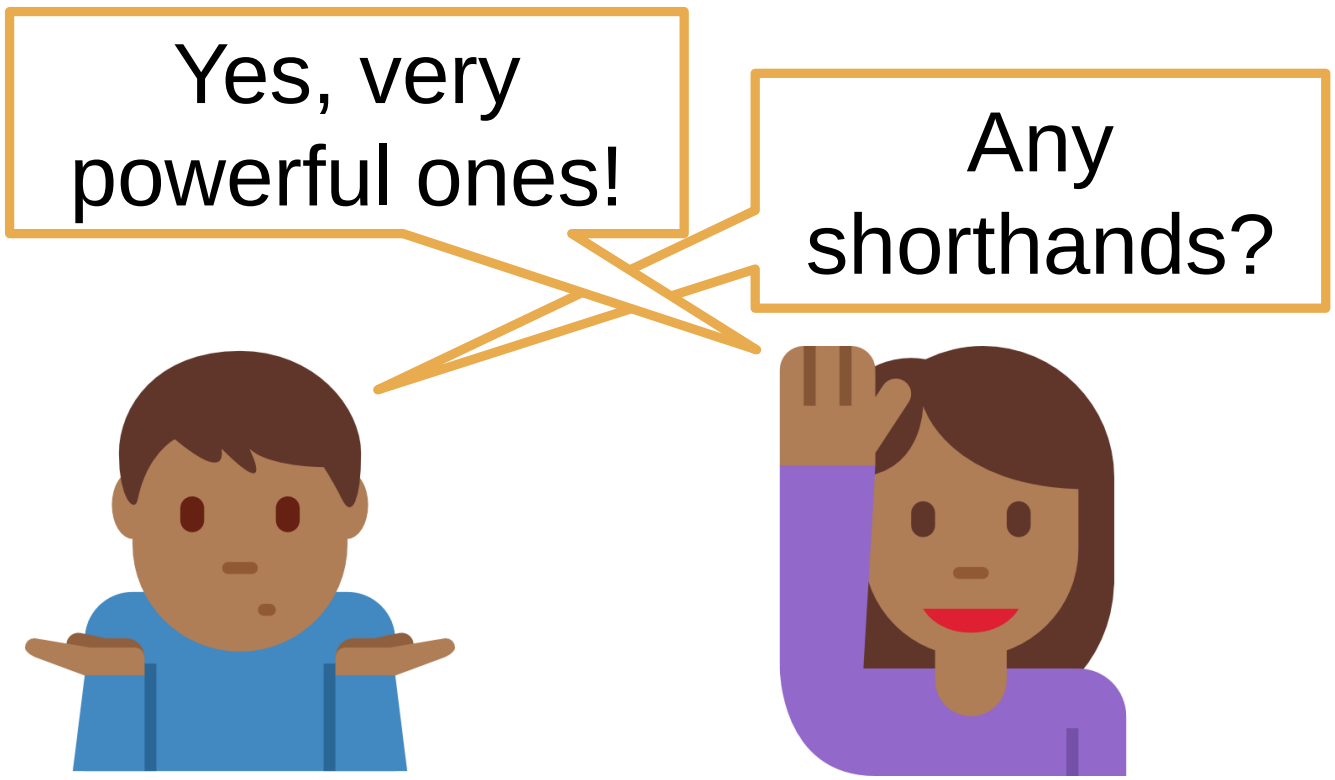

```
#include<stdio.h>
int main(){
int a = 5, b = 4;
printf(“Hello %d %d”,a,b);
return 0;
}
```



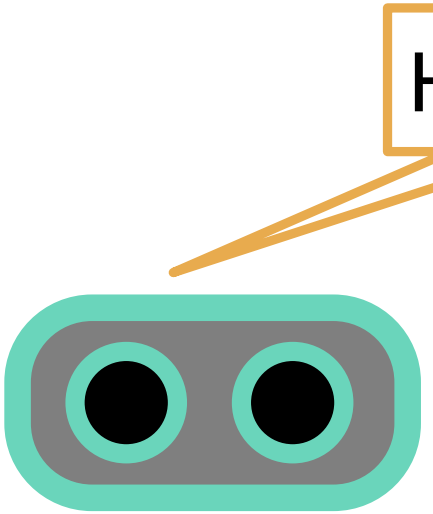
The true power of printf

We have seen how to make Mr. C

- Say things like “Hello”
- Tell us the value of an integer variable




```
#include<stdio.h>
int main(){
int a = 5, b = 4;
printf(“Hello %d %d”,a,b);
return 0;
}
```



The true power of printf

We have seen how to make Mr. C

Say things like "Hell

Tell us the value of

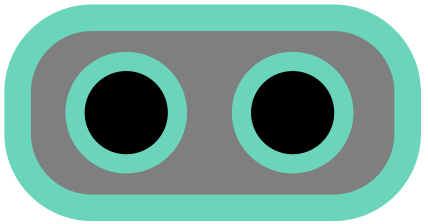
I am confused ☹

Any shorthands?

Yes, very powerful ones!

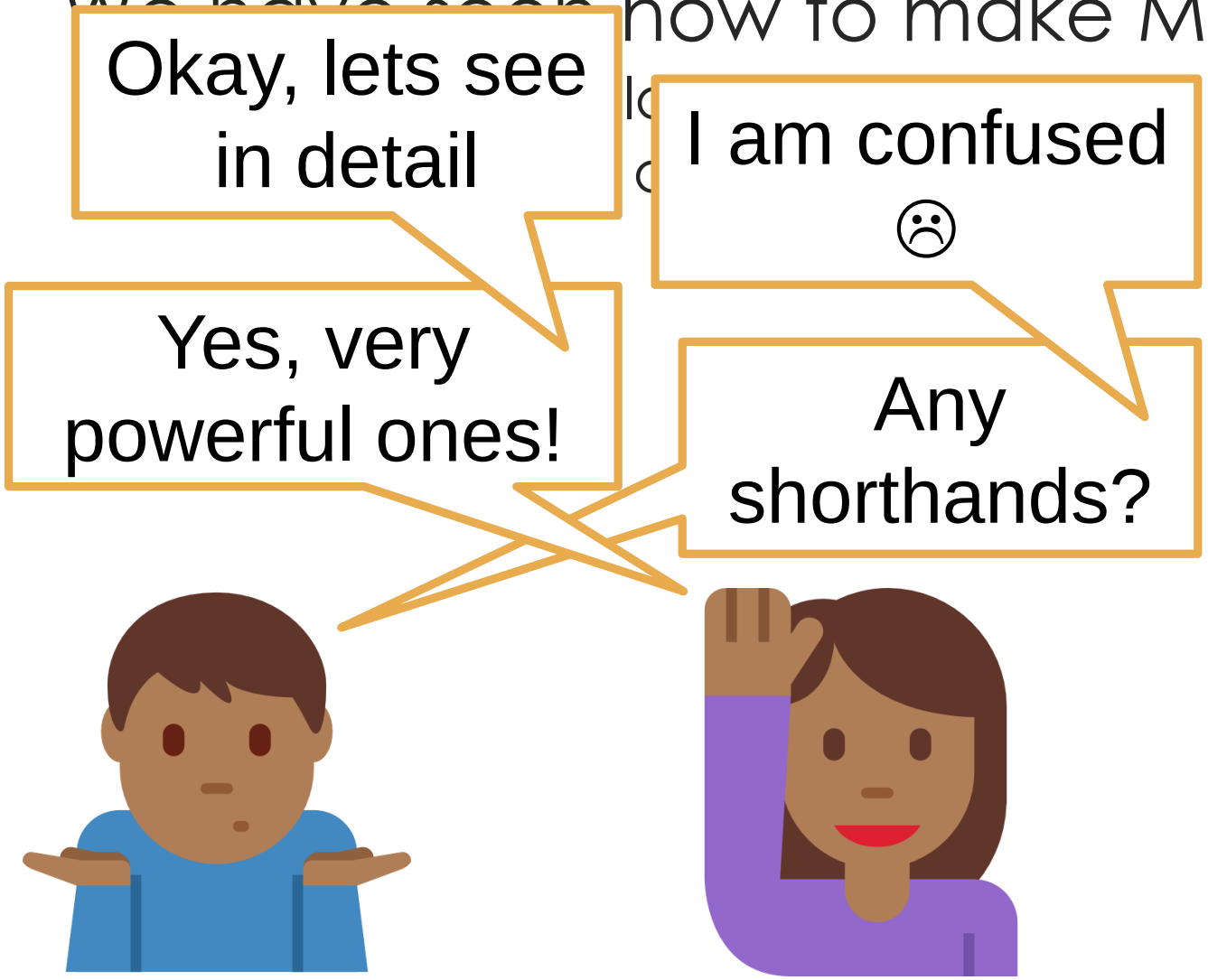
```
#include<stdio.h>
int main(){
int a = 5, b = 4;
printf("Hello %d %d",a,b);
return 0;
}
```

Hello 5 4

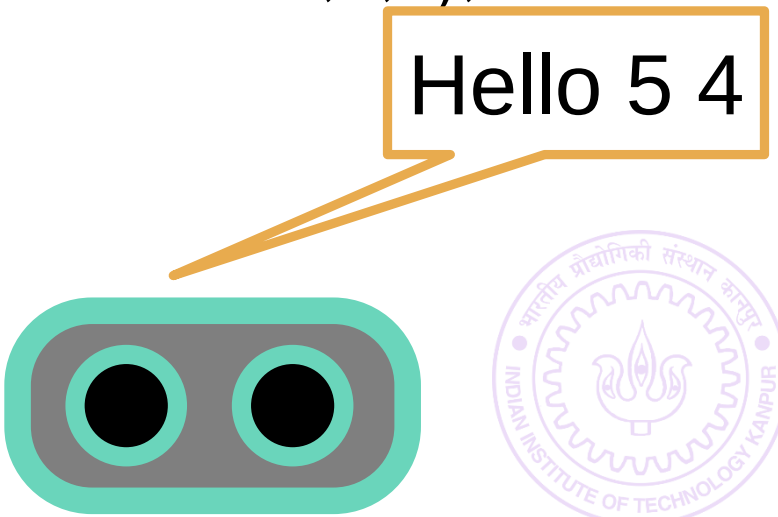


The true power of printf

We have seen how to make Mr. C




```
#include<stdio.h>
int main(){
int a = 5, b = 4;
printf("Hello %d %d",a,b);
return 0;
}
```



The true power of printf

The true power of printf

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

The true power of printf

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

printf follows this exact same rule while telling Mr. C what to print

The true power of printf

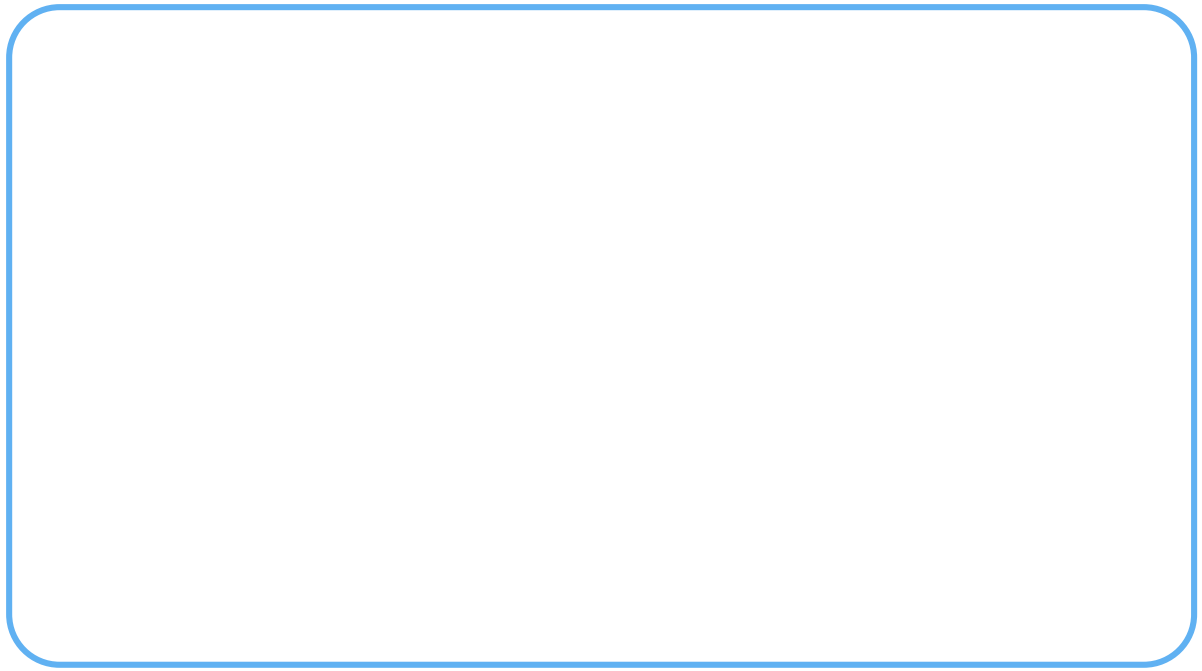
HOW WE USUALLY SPEAK TO A HUMAN

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

printf follows this exact same rule while telling Mr. C what to print

The true power of printf

HOW WE USUALLY SPEAK TO A HUMAN



Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

printf follows this exact same rule while telling Mr. C what to print

The true power of printf

HOW WE USUALLY SPEAK TO A HUMAN

Please write the English word Hello, followed by a space, followed by the value of an integer, followed by a space followed by the value of another integer.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

printf follows this exact same rule while telling Mr. C what to print

The true power of printf

HOW WE USUALLY SPEAK TO A HUMAN

Please write the English word Hello, followed by a space, followed by the value of an integer, followed by a space followed by the value of another integer.

By the way, the first integer to be written is a and the second integer to be written is b.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

printf follows this exact same rule while telling Mr. C what to print

The true power of printf

HOW WE MUST SPEAK TO MR. COMPILER

HOW WE USUALLY SPEAK TO A HUMAN

Please write the English word Hello, followed by a space, followed by the value of an integer, followed by a space followed by the value of another integer.

By the way, the first integer to be written is a and the second integer to be written is b.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

printf follows this exact same rule while telling Mr. C what to print

The true power of printf

HOW WE MUST SPEAK TO MR. COMPILER

HOW WE USUALLY SPEAK TO A HUMAN

Please write the English word Hello, followed by a space, followed by the value of an integer, followed by a space followed by the value of another integer.

By the way, the first integer to be written is a and the second integer to be written is b.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

printf follows this exact same rule while telling Mr. C what to print

The true power of printf

HOW WE MUST SPEAK TO MR. COMPILER

```
printf("Hello %d %d",a,b);
```

HOW WE USUALLY SPEAK TO A HUMAN

Please write the English word Hello, followed by a space, followed by the value of an integer, followed by a space followed by the value of another integer.

By the way, the first integer to be written is a and the second integer to be written is b.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

printf follows this exact same rule while telling Mr. C what to print

The true power of printf

HOW WE MUST SPEAK TO MR. COMPILER

```
printf("Hello %d %d",a,b);
```

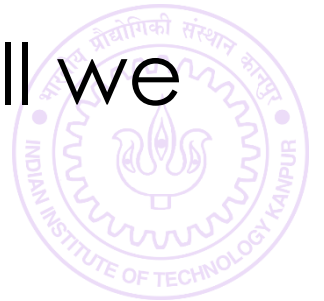
HOW WE USUALLY SPEAK TO A HUMAN

Please write the English word Hello, followed by a space, followed by the value of an integer, followed by a space followed by the value of another integer.

By the way, the first integer to be written is a and the second integer to be written is b.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

printf follows this exact same rule while telling Mr. C what to print



The true power of printf

HOW WE MUST SPEAK TO MR. COMPILER

```
printf("Hello %d %d",a,b);
```

HOW WE USUALLY SPEAK TO A HUMAN

Please write the English word Hello, followed by a space, followed by the value of an integer, followed by a space followed by the value of another integer.

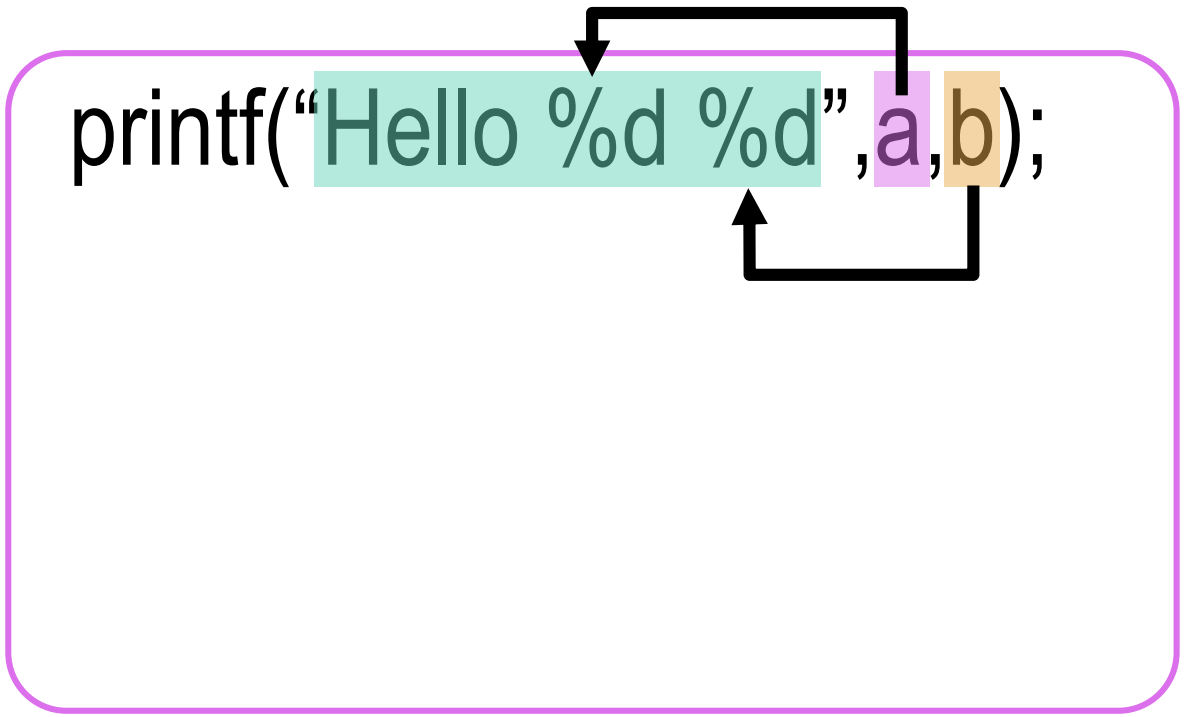
By the way, the first integer to be written is a and the second integer to be written is b.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

printf follows this exact same rule while telling Mr. C what to print

The true power of printf

HOW WE MUST SPEAK TO MR. COMPILER



HOW WE USUALLY SPEAK TO A HUMAN

Please write the English word Hello, followed by a space, followed by the value of an integer, followed by a space followed by the value of another integer.

By the way, the first integer to be written is a and the second integer to be written is b.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

printf follows this exact same rule while telling Mr. C what to print

The true power of printf

HOW WE MUST SPEAK TO MR. COMPILER

```
printf("Hello %d %d",a,b);
```

Format string

HOW WE USUALLY SPEAK TO A HUMAN

Please write the English word Hello, followed by a space, followed by the value of an integer, followed by a space followed by the value of another integer.

By the way, the first integer to be written is a and the second integer to be written is b.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

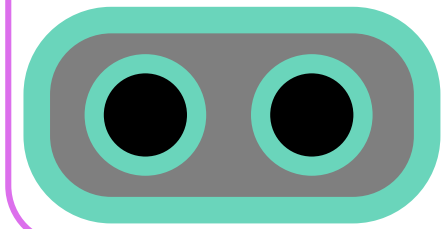
printf follows this exact same rule while telling Mr. C what to print

The true power of printf

HOW WE MUST SPEAK TO MR. COMPILER

```
printf("Hello %d %d",a,b);
```

Format string



HOW WE USUALLY SPEAK TO A HUMAN

Please write the English word Hello, followed by a space, followed by the value of an integer, followed by a space followed by the value of another integer.

By the way, the first integer to be written is a and the second integer to be written is b.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

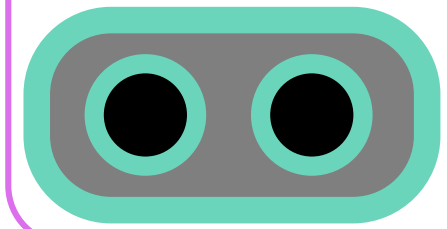
printf follows this exact same rule while telling Mr. C what to print

The true power of printf

HOW WE MUST SPEAK TO MR. COMPILER

printf("Hello %d %d",a,b);

Format string



Format string tells me **how** to print things, and then I am told **what** to print

HOW WE USUALLY SPEAK TO A HUMAN

Please write the English word Hello, followed by a space, followed by the value of an integer, followed by a space followed by the value of another integer.

By the way, the first integer to be written is a and the second integer to be written is b.

Remember Mr. C likes to be told beforehand what all we are going to ask him to do!

printf follows this exact same rule while telling Mr. C what to print

Fun with printf

Fun with printf

Can I print different things on separate lines?

Fun with printf

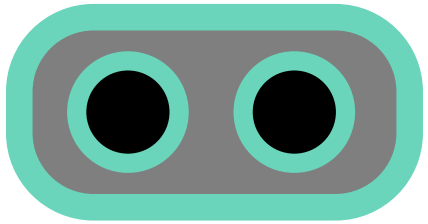
Can I print different things on separate lines?

```
printf("Hello\n%d\n%d",a,b);
```

Fun with printf

Can I print different things on separate lines?

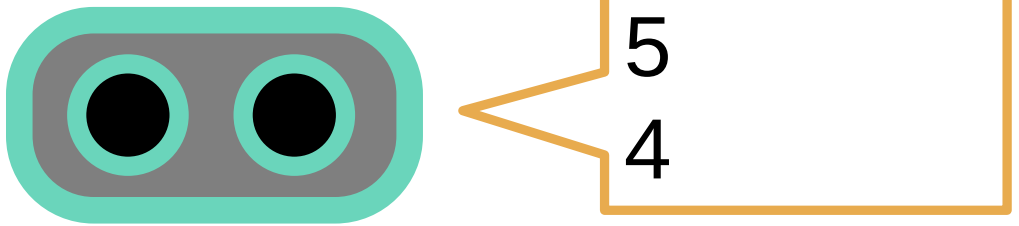
```
printf("Hello\n%d\n%d",a,b);
```



Fun with printf

Can I print different things on separate lines?

```
printf("Hello\n%d\n%d",a,b);
```

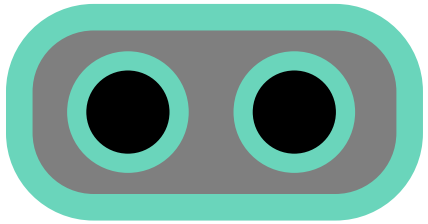


Fun with printf

Can I print different things on separate lines?

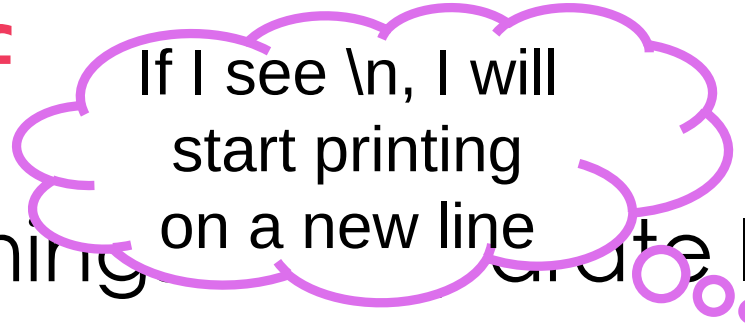
If I see \n, I will start printing on a new line

```
printf("Hello\n%d\n%d",a,b);
```



```
Hello
5
4
```

Fun with printf

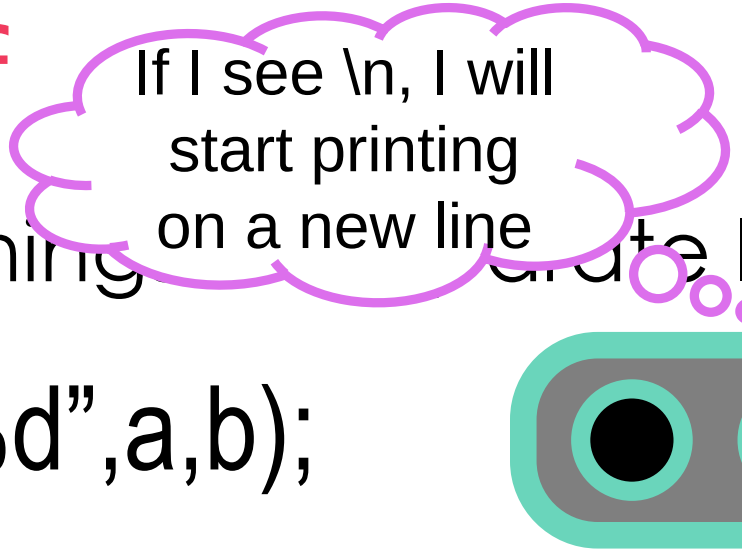


Can I print different things on separate lines?

```
printf("Hello\n%d\n%d",a,b);
```

What if I wish to print the character “ (inverted quotes)?

Fun with printf



Can I print different things on separate lines?

```
printf("Hello\n%d\n%d",a,b);
```



```
Hello
5
4
```

What if I wish to print the character " (inverted quotes)?

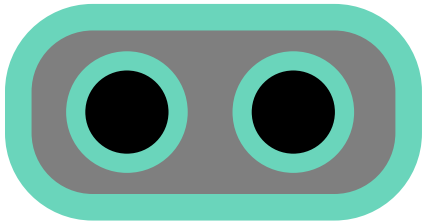
```
printf("\"Hello\"");
```

Fun with printf

If I see \n, I will start printing on a new line

Can I print different things on separate lines?

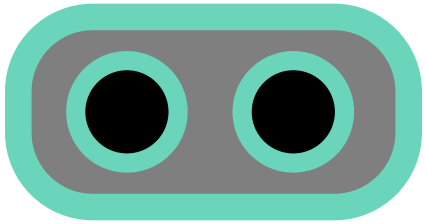
```
printf("Hello\n%d\n%d",a,b);
```



```
Hello
5
4
```

What if I wish to print the character " (inverted quotes)?

```
printf("\"Hello\"");
```

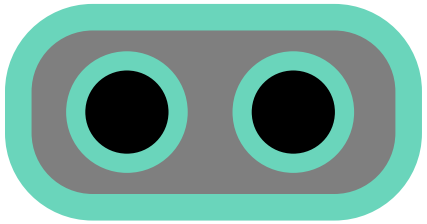


Fun with printf

If I see \n, I will start printing on a new line

Can I print different things on separate lines?

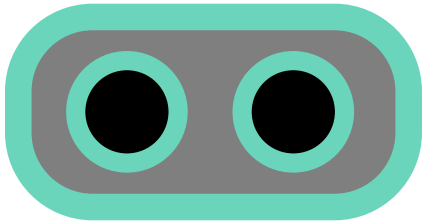
```
printf("Hello\n%d\n%d",a,b);
```



```
Hello
5
4
```

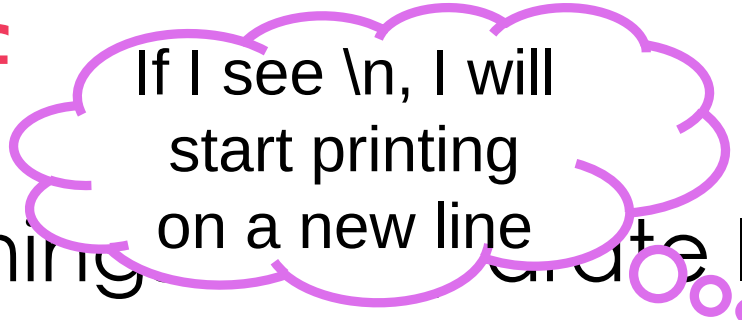
What if I wish to print the character " (inverted quotes)?

```
printf("\"Hello\"");
```



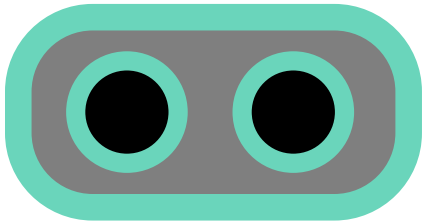
```
"Hello"
```

Fun with printf



Can I print different things on separate lines?

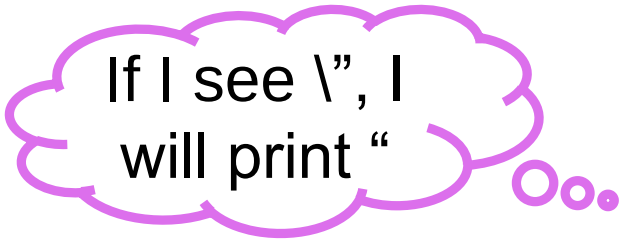
```
printf("Hello\n%d\n%d",a,b);
```

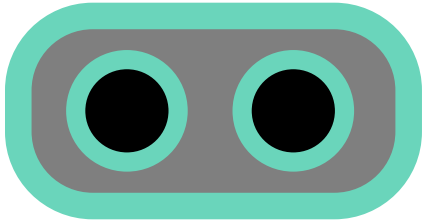


```
Hello
5
4
```

What if I wish to print the character " (inverted quotes)?

```
printf("\"Hello\"");
```





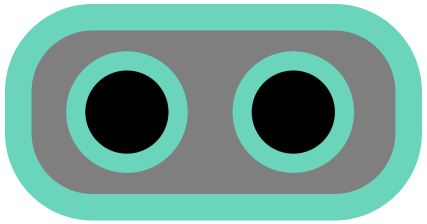
```
"Hello"
```

Fun with printf

If I see \n, I will start printing on a new line

Can I print different things on separate lines?

```
printf("Hello\n%d\n%d",a,b);
```

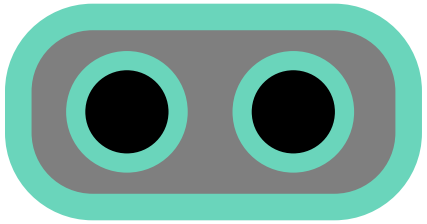


```
Hello
5
4
```

What if I wish to print the character " (inverted quotes)?

If I see \", I will print "

```
printf("\"Hello\"");
```



```
"Hello"
```

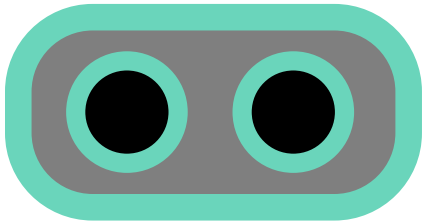
What if I wish to print the character % (percentage sign)?

Fun with printf

If I see \n, I will start printing on a new line

Can I print different things on separate lines?

```
printf("Hello\n%d\n%d",a,b);
```

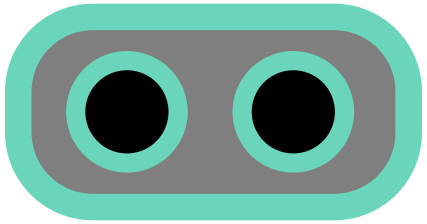


```
Hello
5
4
```

What if I wish to print the character " (inverted quotes)?

If I see \", I will print "

```
printf("\"Hello\"");
```



```
"Hello"
```

What if I wish to print the character % (percentage sign)?

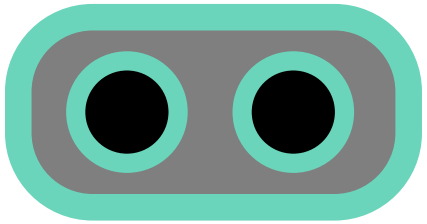
```
printf("90%% marks");
```

Fun with printf

If I see \n, I will start printing on a new line

Can I print different things on separate lines?

```
printf("Hello\n%d\n%d",a,b);
```

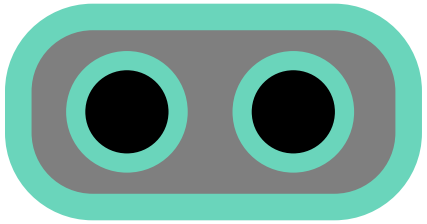


```
Hello
5
4
```

What if I wish to print the character " (inverted quotes)?

If I see \", I will print "

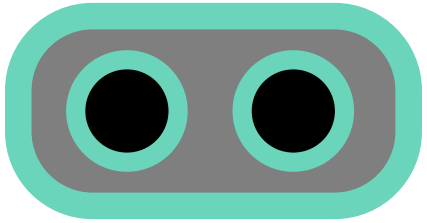
```
printf("\"Hello\"");
```



```
"Hello"
```

What if I wish to print the character % (percentage sign)?

```
printf("90%% marks");
```

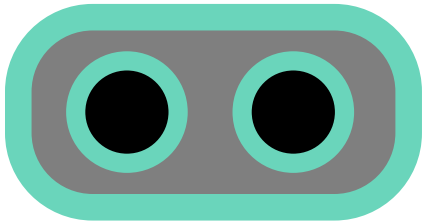


Fun with printf

Can I print different things on separate lines?

If I see \n, I will start printing on a new line

```
printf("Hello\n%d\n%d",a,b);
```

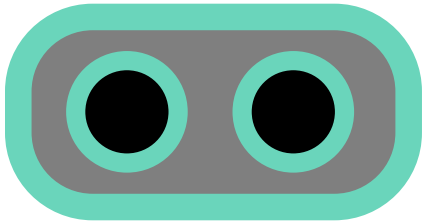


```
Hello
5
4
```

What if I wish to print the character " (inverted quotes)?

```
printf("\"Hello\"");
```

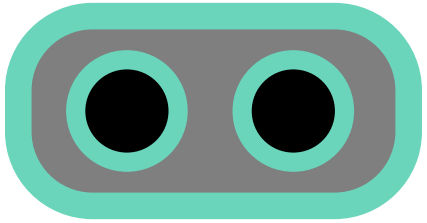
If I see \", I will print "



```
"Hello"
```

What if I wish to print the character % (percentage sign)?

```
printf("90%% marks");
```



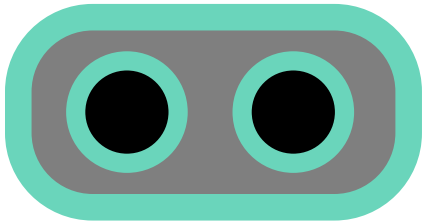
```
90% marks
```

Fun with printf

Can I print different things in separate lines?

If I see \n, I will start printing on a new line

```
printf("Hello\n%d\n%d",a,b);
```

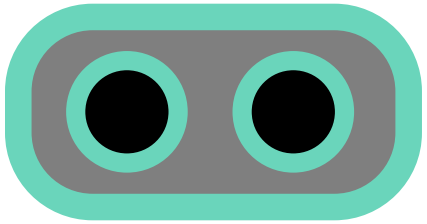


```
Hello
5
4
```

What if I wish to print the character " (inverted quotes)?

```
printf("\"Hello\"");
```

If I see \", I will print "

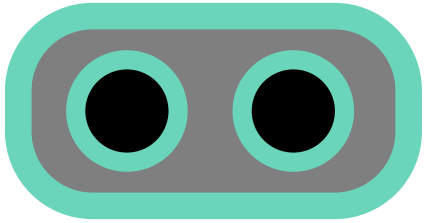


```
"Hello"
```

What if I wish to print the character % (percentage sign)?

```
printf("90%% ma
```

If I see %%, I will print %



```
90% marks
```

Fun with printf

Fun with printf

To print the character \ (backslash)

Fun with printf

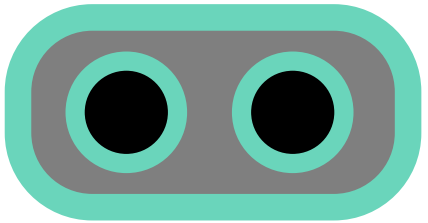
To print the character \ (backslash)

```
printf("To print on new line, use \\n");
```

Fun with printf

To print the character \ (backslash)

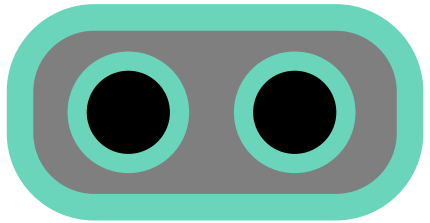
```
printf("To print on new line, use \\n");
```



Fun with printf

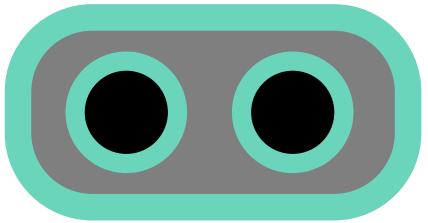
To print the character \ (backslash)

```
printf("To print on new line, use \\n");
```



Fun with printf

To print on new line, use \n

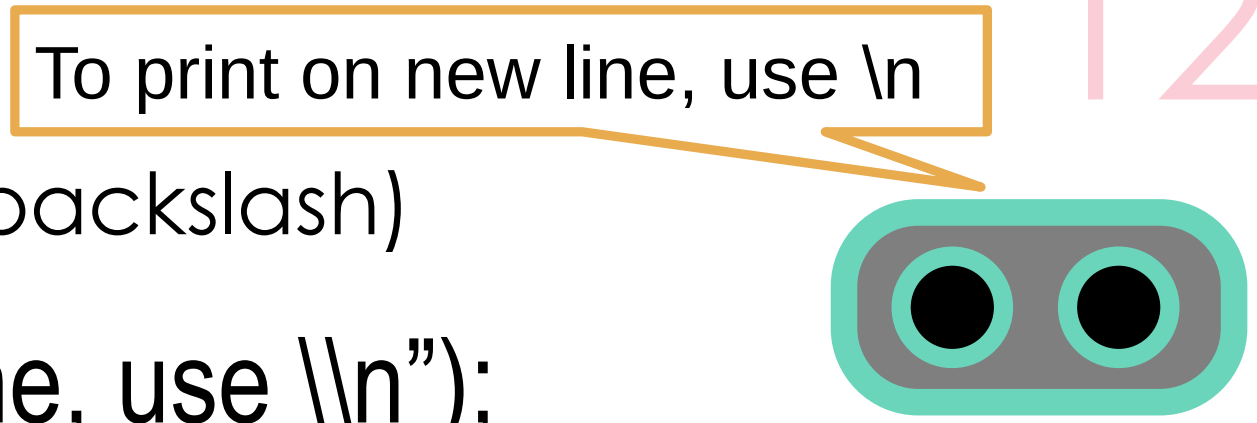


To print the character \ (backslash)

```
printf("To print on new line, use \\n");
```

To print a *tab* character (a longer space)

Fun with printf



To print the character \ (backslash)

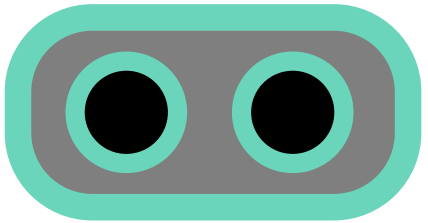
```
printf("To print on new line, use \\n");
```

To print a *tab* character (a longer space)

```
printf("Very\tNice");
```

Fun with printf

To print on new line, use \n

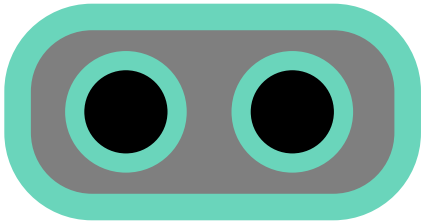


To print the character \ (backslash)

```
printf("To print on new line, use \\n");
```

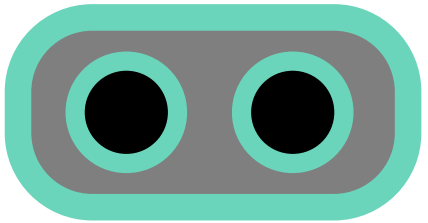
To print a *tab* character (a longer space)

```
printf("Very\tNice");
```



Fun with printf

To print on new line, use \n

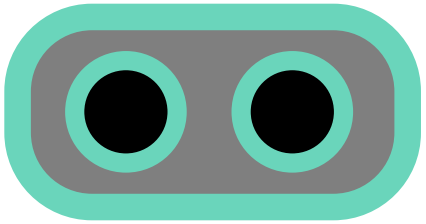


To print the character \ (backslash)

```
printf("To print on new line, use \\n");
```

To print a *tab* character (a longer space)

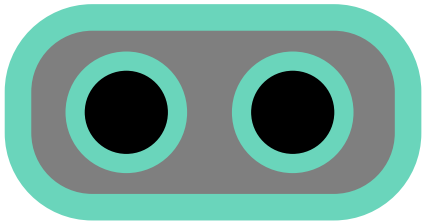
```
printf("Very\tNice");
```



Very Nice

Fun with printf

To print on new line, use \n

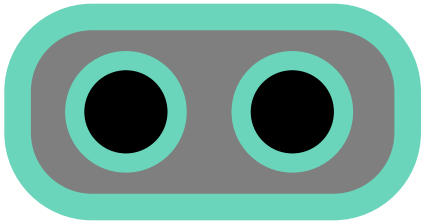


To print the character \ (backslash)

```
printf("To print on new line, use \\n");
```

To print a *tab* character (a longer space)

```
printf("Very\tNice");
```

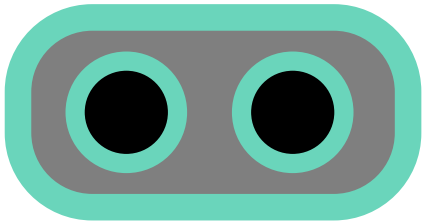


Very Nice

Allows us to print very nicely formatted output ☺

Fun with printf

To print on new line, use \n

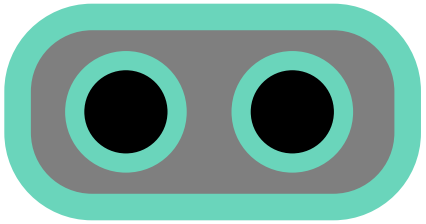


To print the character \ (backslash)

```
printf("To print on new line, use \\n");
```

To print a *tab* character (a longer space)

```
printf("Very\tNice");
```



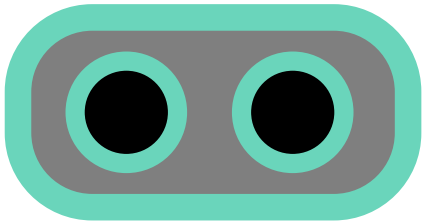
Very Nice

Allows us to print very nicely formatted output ☺

More examples in labs – till then, have fun on your own

Fun with printf

To print on new line, use \n

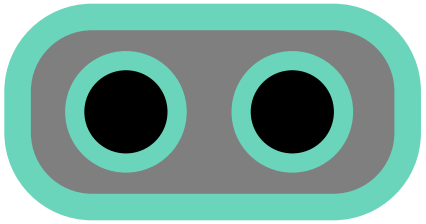


To print the character \ (backslash)

```
printf("To print on new line, use \\n");
```

To print a *tab* character (a longer space)

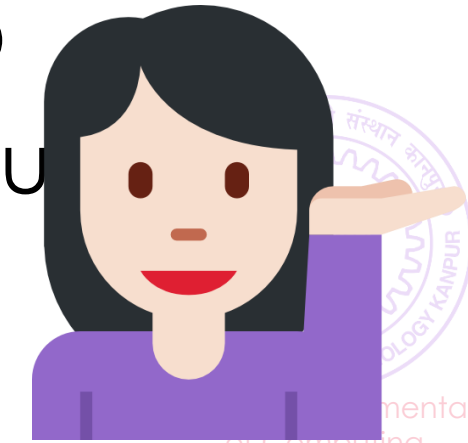
```
printf("Very\tNice");
```



Very Nice

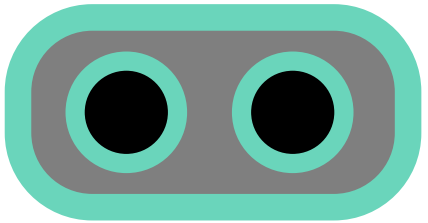
Allows us to print very nicely formatted output ☺

More examples in labs – till then, have fun on you



Fun with printf

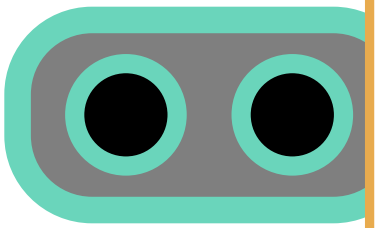
To print on new line, use \n



To print the character \ (backslash)

```
printf("To print on new line, use \\n");
```

To print a *tab* character (a lo



```
printf("Very\tNice");
```

\n, \" called *escape sequences* since they "escape" the normal rules

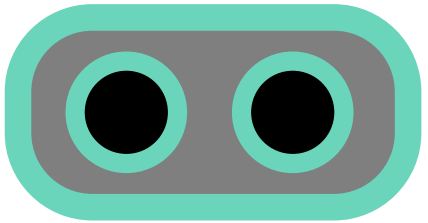
Allows us to print very nicely formatted output ☺

More examples in labs – till then, have fun on you

Fun with printf

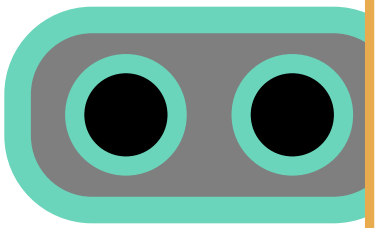
To print on new line, use \n

To print the character \ (backslash)



```
printf("To print on new line, use \\n");
```

To print a *tab* character (a lo



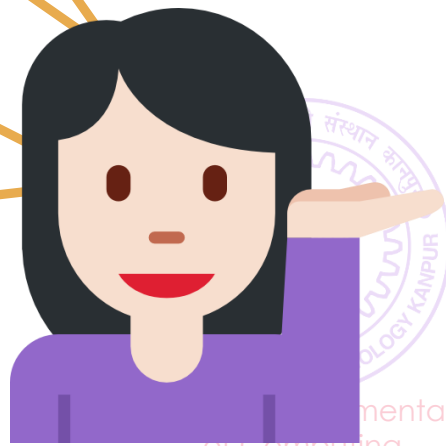
```
printf("Very\tNice");
```

\n, \" called *escape sequences* since they "escape" the normal rules

Allows us to print very nicely formatted output ☺

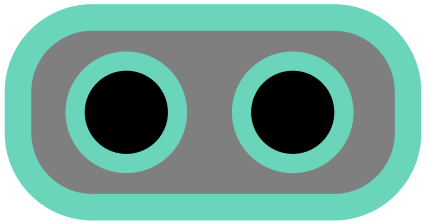
More examples in labs – till th

Experiment with them to get comfortable



Fun with printf

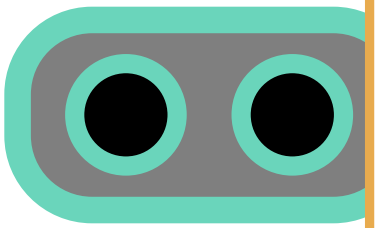
To print on new line, use \n



To print the character \ (backslash)

```
printf("To print on new line, use \\n");
```

To print a *tab* character (a lo



```
printf("Very\tNice");
```

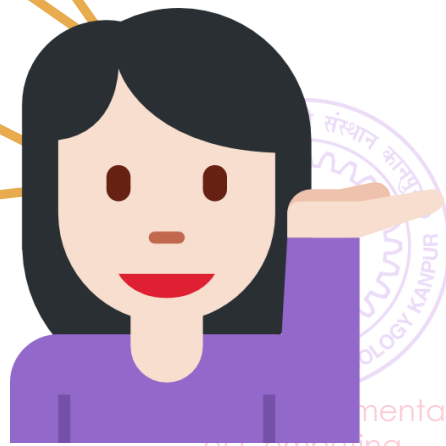
\n, \" called *escape sequences* since they "escape" the normal rules

Allows us to print very nicely formatted output ☺

More e

Need to be careful, especially with things like printf("\"Hello\"");

Experiment with them to get comfortable



Fun with Integers

Fun with Integers

Operation	C Code	a	b	c
Addition	c = a + b;	5	4	9
Subtraction	c = a – b;	4	5	-1
Multiplication	c = a * b;	-2	-4	8
Division	c = a / b;	7	2	3
Remainder	c = a % b;	7	2	1

Fun with Integers

Operation	C Code	a	b	c
Addition	c = a + b;	5	4	9
Subtraction	c = a – b;	4	5	-1
Multiplication	c = a * b;	-2	-4	8
Division	c = a / b;	7	2	3
Remainder	c = a % b;	7	2	1

Be careful: in math we often write z = 2xy

Fun with Integers

Operation	C Code	a	b	c
Addition	c = a + b;	5	4	9
Subtraction	c = a – b;	4	5	-1
Multiplication	c = a * b;	-2	-4	8
Division	c = a / b;	7	2	3
Remainder	c = a % b;	7	2	1

Be careful: in math we often write z = 2xy

Mr C will not like it.

Fun with Integers

Operation	C Code	a	b	c
Addition	c = a + b;	5	4	9
Subtraction	c = a – b;	4	5	-1
Multiplication	c = a * b;	-2	-4	8
Division	c = a / b;	7	2	3
Remainder	c = a % b;	7	2	1

Be careful: in math we often write z = 2xy

- Mr C will not like it.
- He will want z = 2 * x * y;

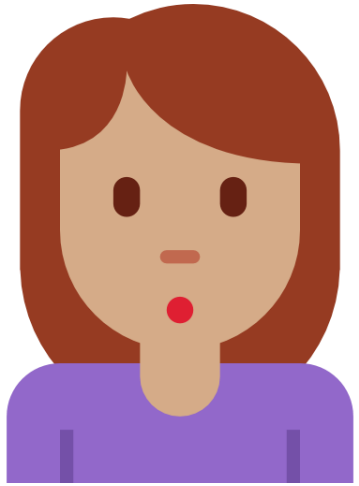
Fun with Integers

Operation	C Code	a	b	c
Addition	c = a + b;	5	4	9
Subtraction	c = a – b;	4	5	-1
Multiplication	c = a * b;	-2	-4	8
Division	c = a / b;	7	2	3
Remainder	c = a % b;	7	2	1

Be careful: in math we often write z = 2xy

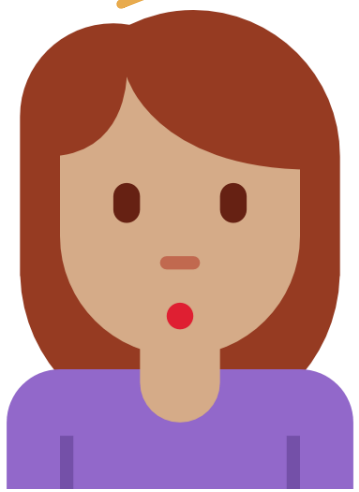
Mr C will not like it.

He will want z = 2 * x * y;



Fun with Integers

Operation	C Code	a	b	c
Addition	c = a + b;	5	4	9
Subtraction	c = a – b;	4	5	-1
Multiplicatio			-4	8
Division			2	3
Remai…der			2	1



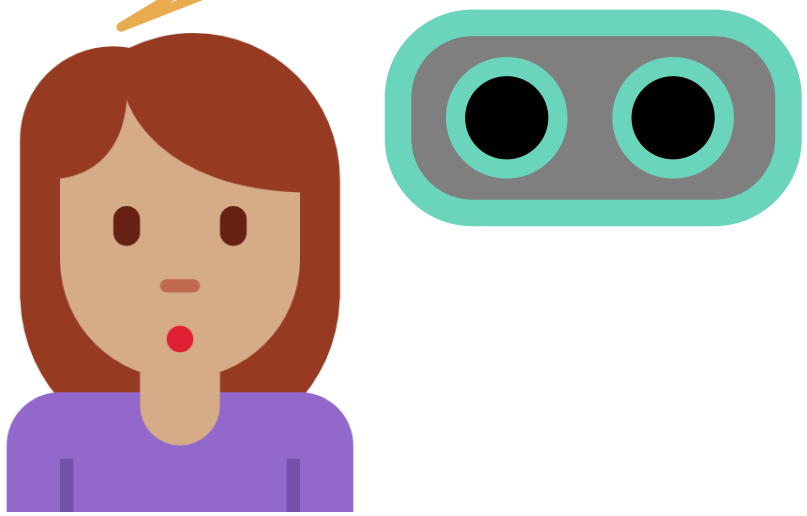
Oh! So Mr. C allows us to use constants too?

Be careful: in math we often write z = 2xy

- Mr C will not like it.
- He will want z = 2 * x * y;

Fun with Integers

Operation	C Code	a	b	c
Addition	c = a + b;	5	4	9
Subtraction	c = a – b;	4	5	-1
Multiplicatio			-4	8
Division			2	3
Remai…der			2	1



Be careful: in math we often write z = 2xy

- Mr C will not like it.
- He will want z = 2 * x * y;

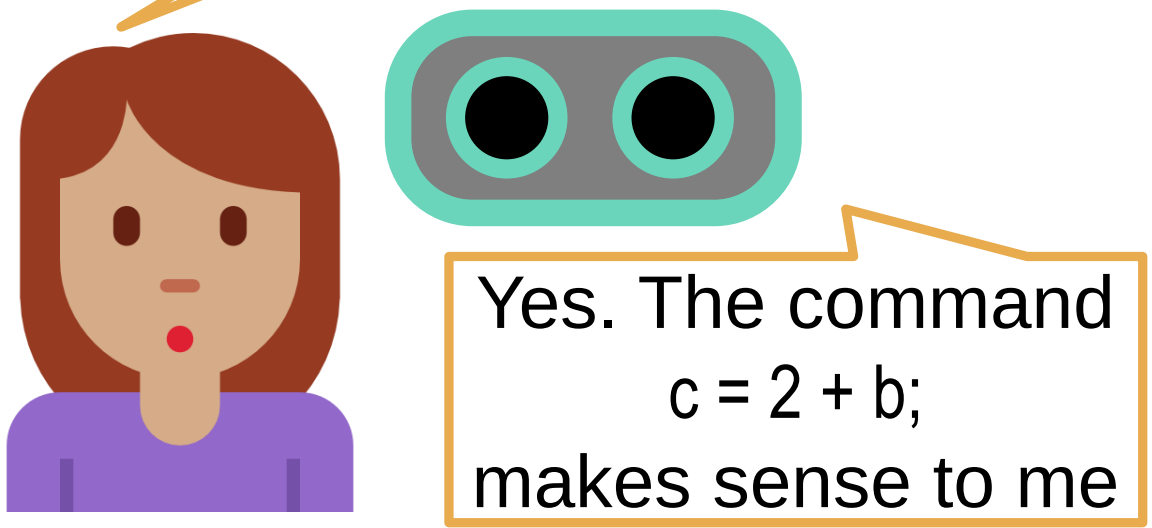
Fun with Integers

Operation	C Code	a	b	c
Addition	c = a + b;	5	4	9
Subtraction	c = a – b;	4	5	-1
Multiplicatio			-4	8
Division			2	3
Remai…der			2	1

Be careful: in math we often write z = 2xy

Mr C will not like it.

He will want z = 2 * x * y;



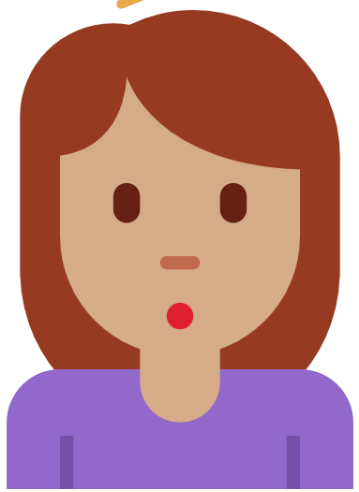
Fun with Integers

Operation	C Code	a	b	c
Addition	c = a + b;	5	4	9
Subtraction	c = a – b;	4	5	-1
Multiplicatio			-4	8
Division			2	3
Rema der			2	1

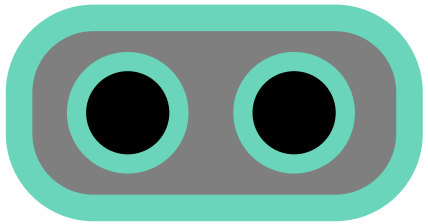
Be careful: in math we often write z = 2xy

Mr C will not like it.

He will want z = 2 * x * y;



Oh! So Mr. C allows us to use constants too?



Not everything needs to be stored in a variable

Yes. The command
c = 2 + b;
makes sense to me

Fun with Integers

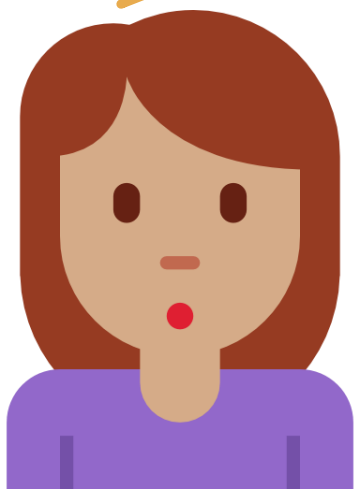
Operation	C Code	a	b	c
Addition	c = a + b;	5	4	9
Subtraction	c = a – b;	4	5	-1
Multiplicatio			-4	8
Division			2	3
Rema der			2	1

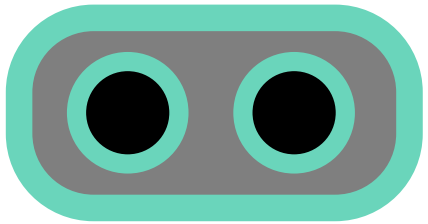
Be careful: in math we often write z = 2xy

Mr C will not like it.

He will want z = 2 * x * y;

Oh! So Mr. C allows us to use constants too?





Not everything needs to be stored in a variable

Yes. The command

c = 2 + b;

makes sense to me

printf("%d %d",a,10);

is fine too ☺

Fun with Integers

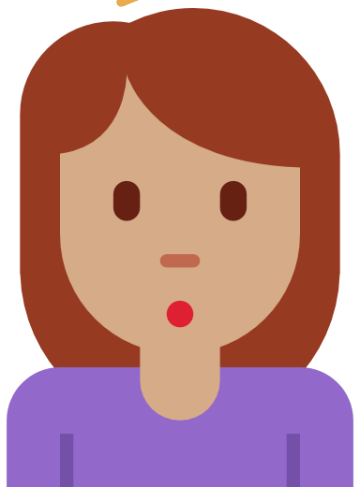
Operation	C Code	a	b	c
Addition	c = a + b;	5	4	9
Subtraction	c = a – b;	4	5	-1
Multiplicatio			-4	8
Division			2	3
Remai…der			2	1

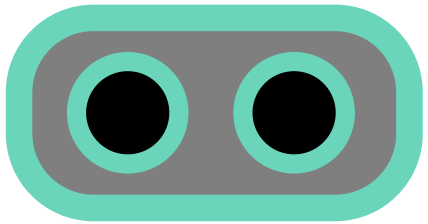
Be careful: in math we often write z = 2xy

Mr C will not like it.

He will want z = 2 * x * y;

Oh! So Mr. C allows us to use constants too?





Not everything needs to be stored in a variable

Yes. The command

c = 2 + b;

makes sense to me

printf("%d %d",a,10);

is fine too ☺

Come to the lab and give it a try!

Fun with Integers

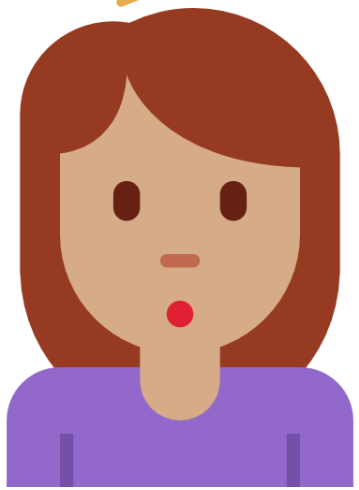
Operation	C Code	a	b	c
Addition	c = a + b;	5	4	9
Subtraction	c = a – b;	4	5	-1
Multiplicatio			-4	8
Division			2	3
Rema der			2	1

Be careful: in math we often write z = 2xy

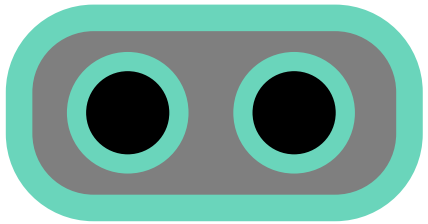
Mr C will not like it.

He will want z = 2 * x * y;

Also be careful about division and remainder



Oh! So Mr. C allows us to use constants too?



Not everything needs to be stored in a variable

Yes. The command c = 2 + b; makes sense to me

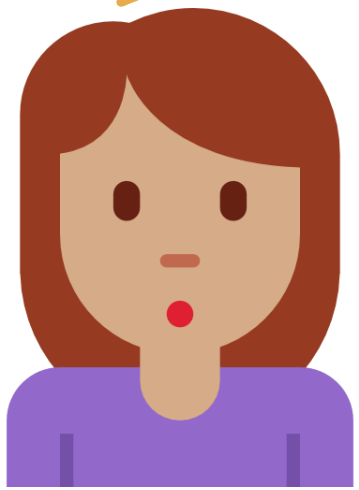
printf("%d %d",a,10); is fine too ☺

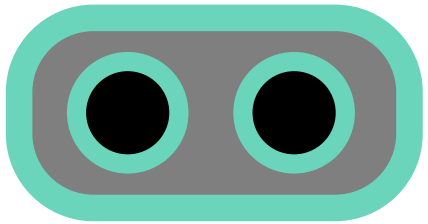
Come to the lab and give it a try!

Fun with Integers

Operation	C Code	a	b	c
Addition	c = a + b;	5	4	9
Subtraction	c = a – b;	4	5	-1
Multiplicatio			-4	8
Division			2	3
Rema…der			2	1

Oh! So Mr. C allows us to use constants too?





Not everything needs to be stored in a variable

Yes. The command
c = 2 + b;
makes sense to me

printf("%d %d",a,10);
is fine too ☺

Come to the lab and give it a try!

Be careful: in math we often write z = 2xy

Mr C will not like it.

He will want z = 2 * x * y;

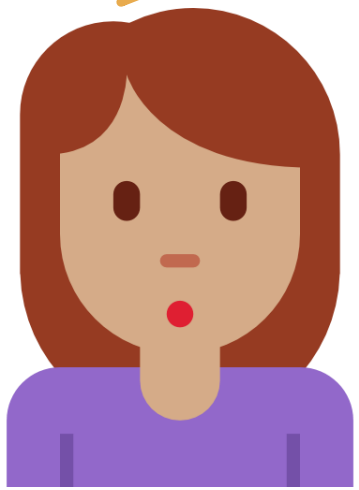
Also be careful about division and remainder

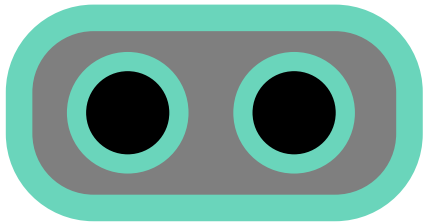
7 / 2 is actually 3.5 but since c is an integer variable, it just stores 3. Remainder is 1

Fun with Integers

Operation	C Code	a	b	c
Addition	c = a + b;	5	4	9
Subtraction	c = a – b;	4	5	-1
Multiplicatio			-4	8
Division			2	3
Rema der			2	1

Oh! So Mr. C allows us to use constants too?





Not everything needs to be stored in a variable

Yes. The command
c = 2 + b;
makes sense to me

printf("%d %d",a,10);
is fine too ☺

Come to the lab and give it a try!

Be careful: in math we often write z = 2xy

- Mr C will not like it.
- He will want z = 2 * x * y;

Also be careful about division and remainder

- 7 / 2 is actually 3.5 but since c is an integer variable, it just stores 3. Remainder is 1

Experiment on your own – will revisit these very soon

Beyond Integers

Beyond Integers

- Lots of fun possible with integers alone

Beyond Integers

- Lots of fun possible with integers alone
  - However, the box storing integers is actually not very big

Beyond Integers

- Lots of fun possible with integers alone
  - However, the box storing integers is actually not very big

Beyond Integers

- Lots of fun possible with integers alone
  - However, the box storing integers is actually not very big
  - Can only store integers between -2,147,483,648 and 2,147,483,647

Beyond Integers

- Lots of fun possible with integers alone
  - However, the box storing integers is actually not very big
  - Can only store integers between -2,147,483,648 and 2,147,483,647
- Also, what about real numbers (fractions etc)

Beyond Integers

- Lots of fun possible with integers alone
  - However, the box storing integers is actually not very big
  - Can only store integers between -2,147,483,648 and 2,147,483,647
- Also, what about real numbers (fractions etc)
  - How to ask Mr C to work with a real number?

Beyond Integers

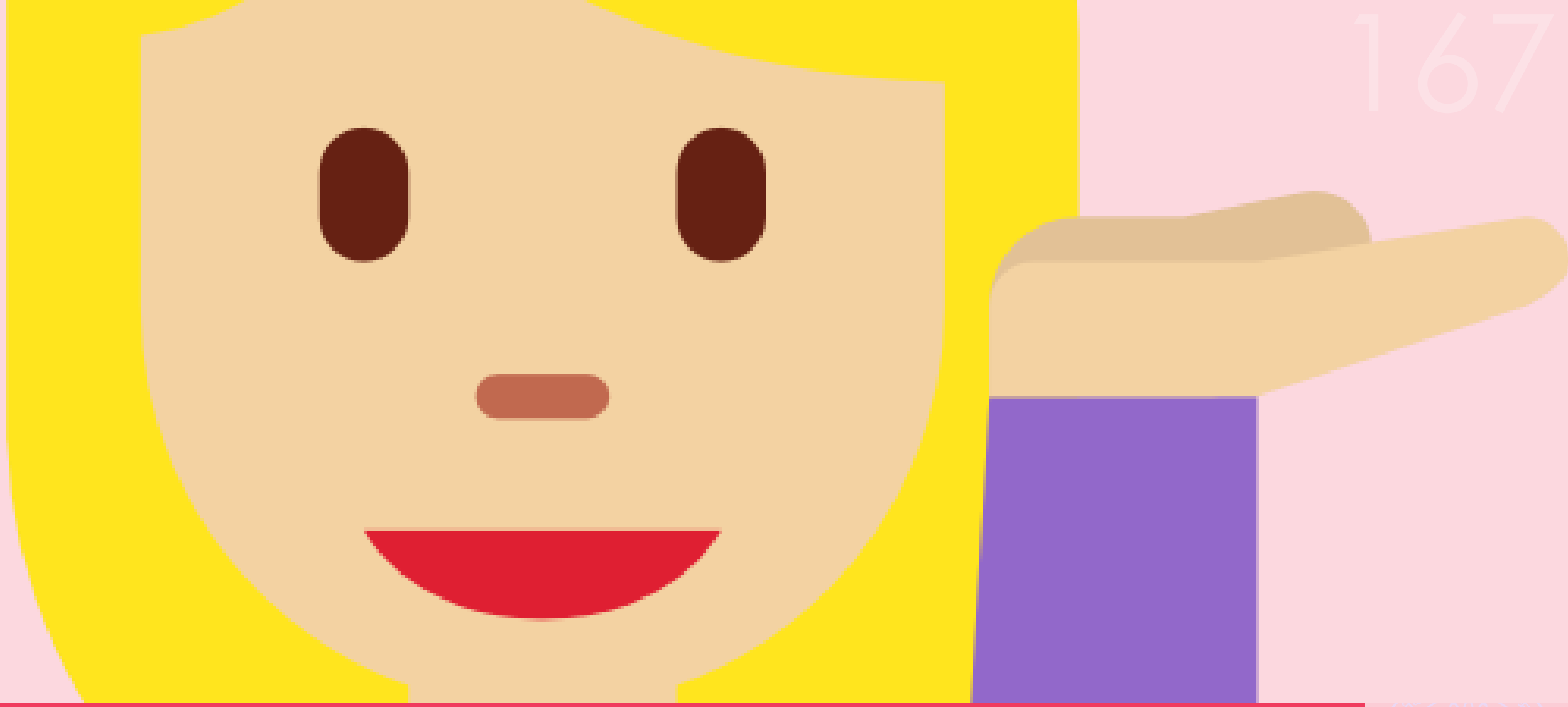
- Lots of fun possible with integers alone
  - However, the box storing integers is actually not very big
  - Can only store integers between -2,147,483,648 and 2,147,483,647
- Also, what about real numbers (fractions etc)
  - How to ask Mr C to work with a real number?
  - How to ask Mr C to print a real number?

Beyond Integers

- Lots of fun possible with integers alone
  - However, the box storing integers is actually not very big
  - Can only store integers between -2,147,483,648 and 2,147,483,647
- Also, what about real numbers (fractions etc)
  - How to ask Mr C to work with a real number?
  - How to ask Mr C to print a real number?
- Next class: long, float, double, long double

Beyond Integers

- Lots of fun possible with integers alone
  - However, the box storing integers is actually not very big
  - Can only store integers between -2,147,483,648 and 2,147,483,647
- Also, what about real numbers (fractions etc)
  - How to ask Mr C to work with a real number?
  - How to ask Mr C to print a real number?
- Next class: long, float, double, long double
- C designers were really nice with names ☺



Emoticons from Flaticon, designed by Twitter
https://www.flaticon.com/packs/smileys-and-people-9