Mr C has a long floating name

ESC101: Foundations of Computing

Purushottam Kar

Announcements

- Special session on using computers
  - August 11, 2018 (coming Saturday), 5PM, NCL CC-02
  - Not a revision class – only for students who are new to computers
  - Thanks to Parth Sharma (tutor) for the suggestion and offer

Adding two unknown numbers

Adding two unknown numbers

HOW WE USUALLY SPEAK TO A HUMAN

Adding two unknown numbers

HOW WE USUALLY SPEAK TO A HUMAN

Adding two unknown numbers

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Adding two unknown numbers

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello

Adding two unknown numbers

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.

Adding two unknown numbers

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

Adding two unknown numbers

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

Please ask me for value of b.

Adding two unknown numbers

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

Please ask me for value of b.

Please print their sum.

Adding two unknown numbers

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

Please ask me for value of b.

Please print their sum.

Goodbye

Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye

Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

Please ask me for value of b.

Please print their sum.

Goodbye

Adding two unknown 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 are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye

Adding two unknown 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 are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye

Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

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

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye

Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
int a, b;
scanf("%d", &a);
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye

Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
int a, b;
scanf("%d", &a);
scanf("%d", &b);
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye

Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
int a, b;
scanf("%d", &a);
scanf("%d", &b);
printf("%d", a + b);
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye

Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

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

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye

Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

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

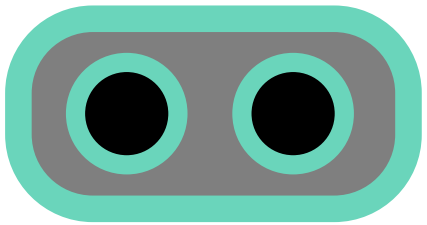
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye

Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

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



HOW WE USUALLY SPEAK TO A HUMAN

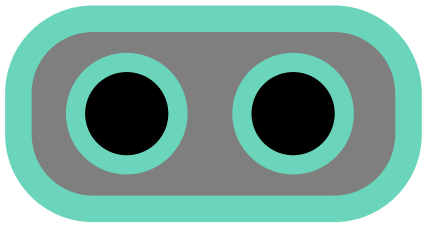
Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye

Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

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

a



HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

Please ask me for value of b.

Please print their sum.

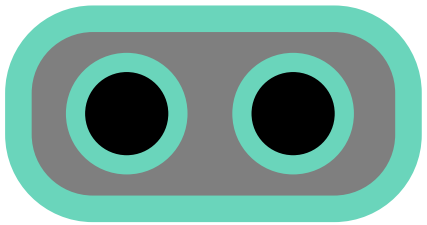
Goodbye

Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

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

a b



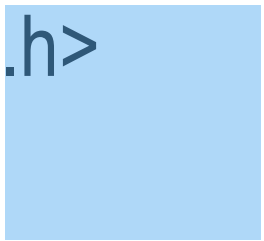
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye

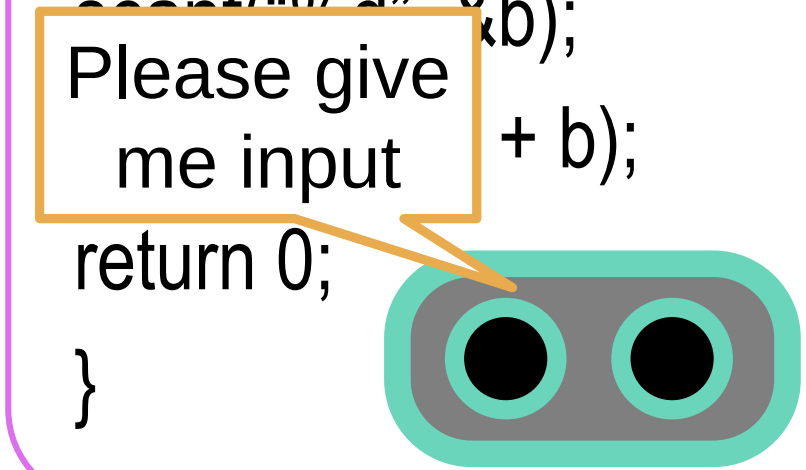
Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

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



a b



HOW WE USUALLY SPEAK TO A HUMAN

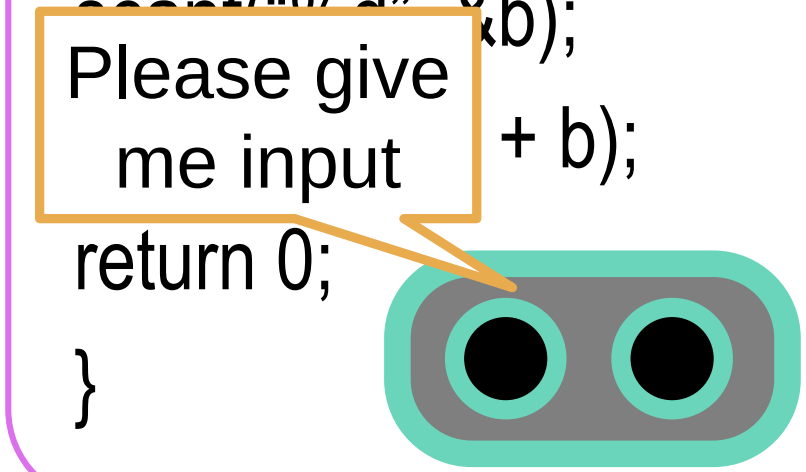
Do you speak English?
Hello
a, b are two variables.
Please ask me for value of a.
Please ask me for value of b.
Please print their sum.
Goodbye

Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

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

a b



HOW WE USUALLY SPEAK TO A HUMAN

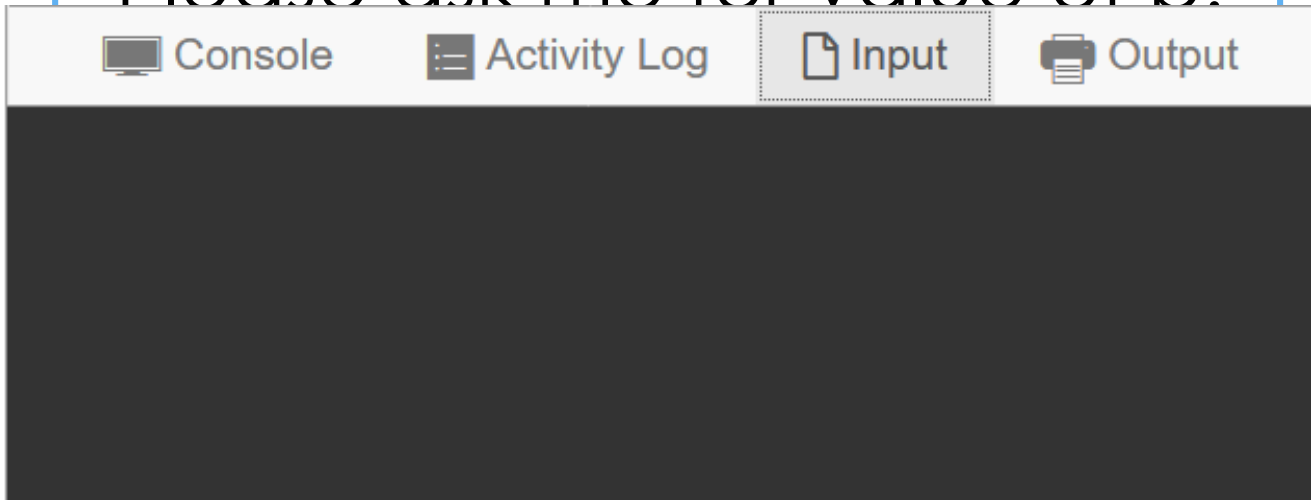
Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

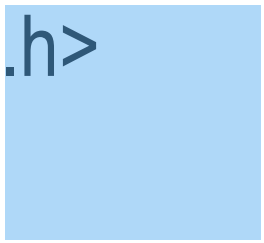
Please ask me for value of b.



Adding two unknown numbers

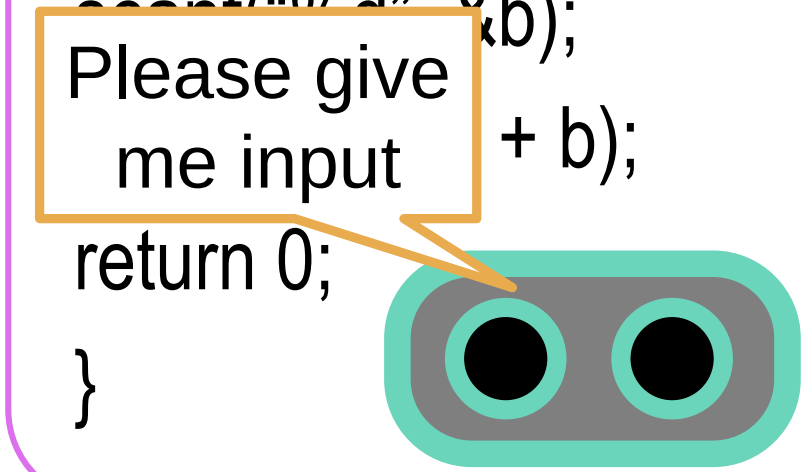
HOW WE MUST SPEAK TO MR. COMPILER

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



a

b



HOW WE USUALLY SPEAK TO A HUMAN

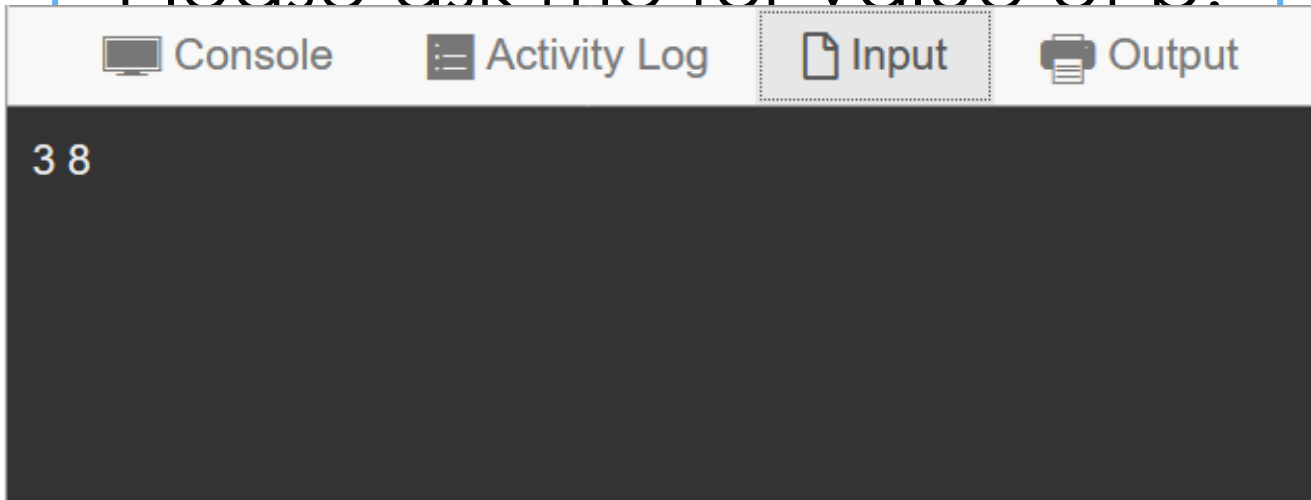
Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

Please ask me for value of b.



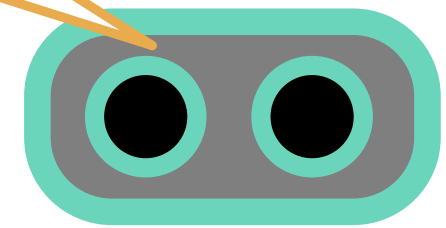
Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
                                  b
                    &b);
                    + b);
return 0;
}
```

Thanks. Let me get back to work

Please give me input



HOW WE USUALLY SPEAK TO A HUMAN

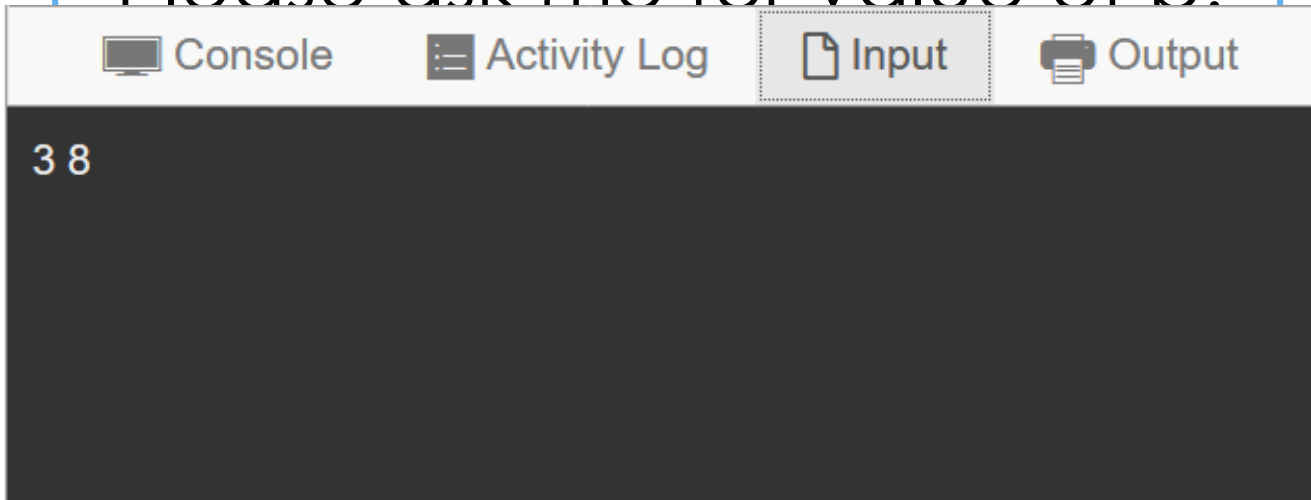
Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

Please ask me for value of b.

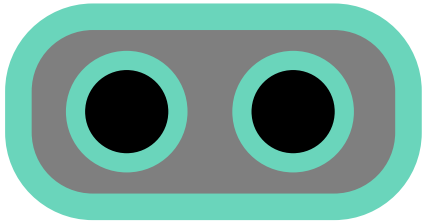


Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

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

a b



HOW WE USUALLY SPEAK TO A HUMAN

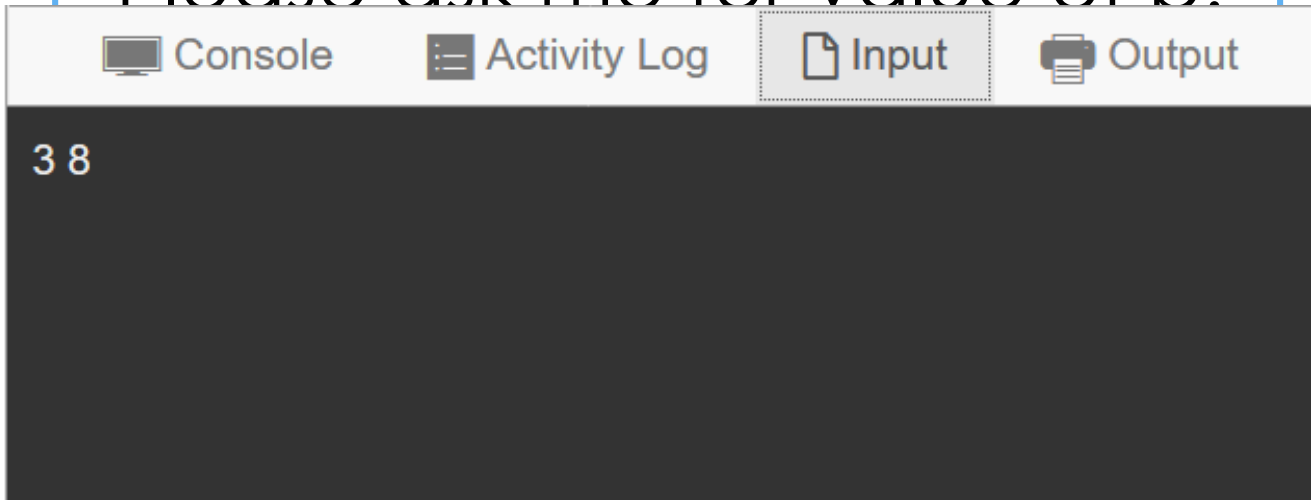
Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

Please ask me for value of b.



Adding two unknown numbers

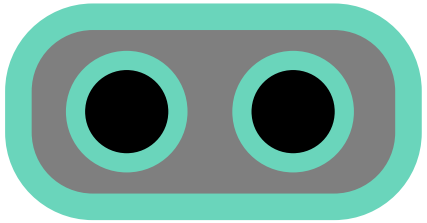
HOW WE MUST SPEAK TO MR. COMPILER

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

3

a

b



HOW WE USUALLY SPEAK TO A HUMAN

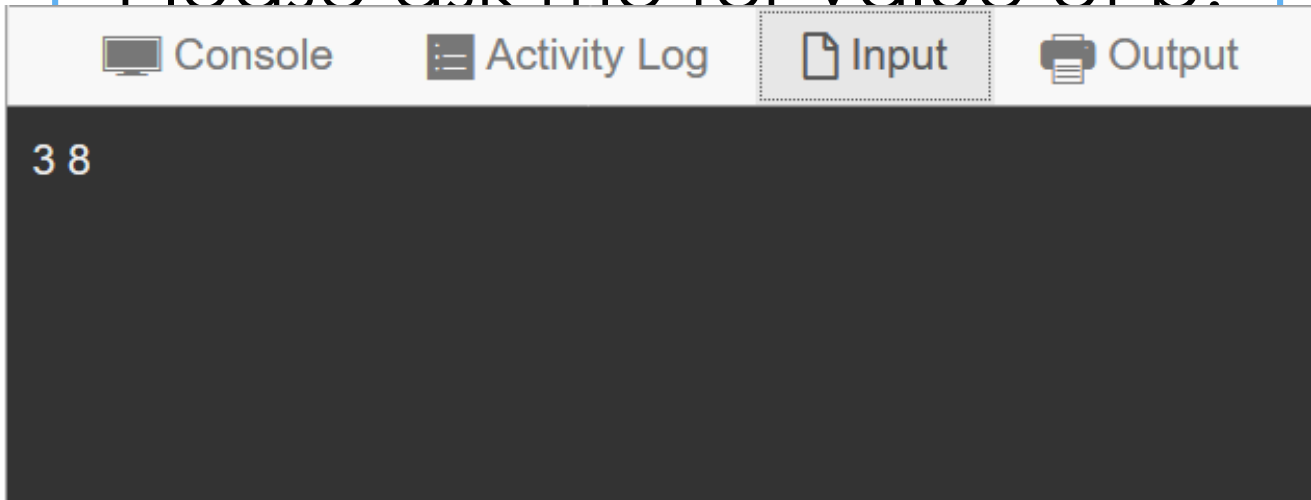
Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

Please ask me for value of b.



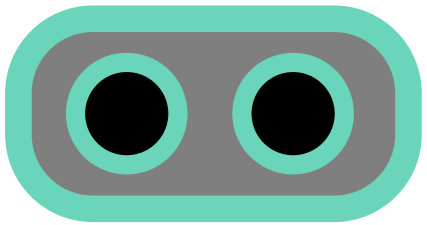
Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

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

3 a

8 b



HOW WE USUALLY SPEAK TO A HUMAN

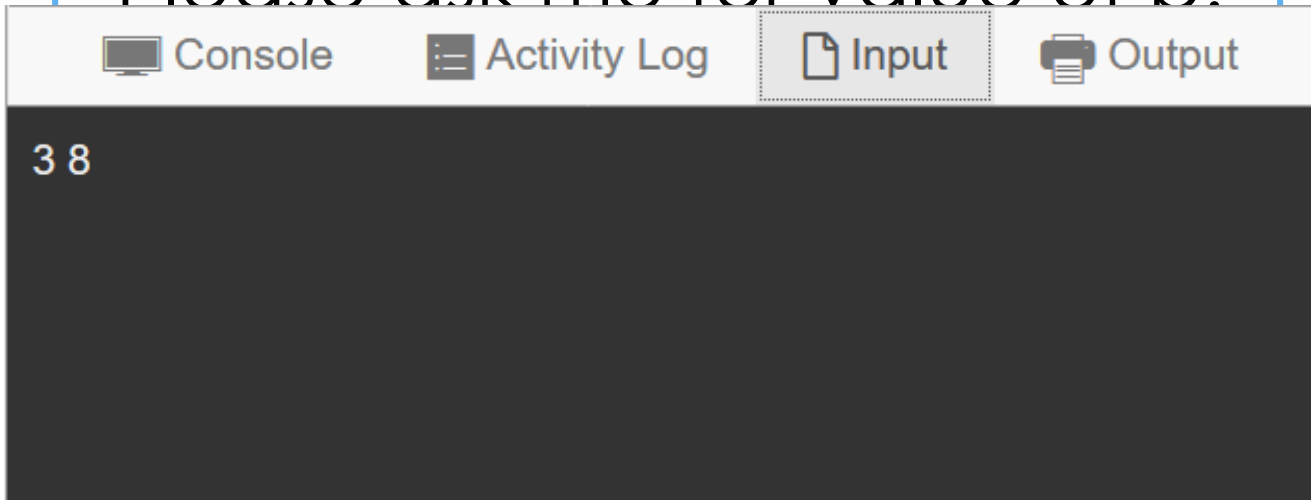
Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

Please ask me for value of b.

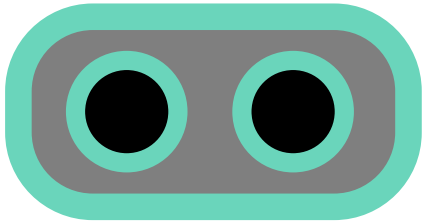


Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

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

3 a 8 b 11



HOW WE USUALLY SPEAK TO A HUMAN

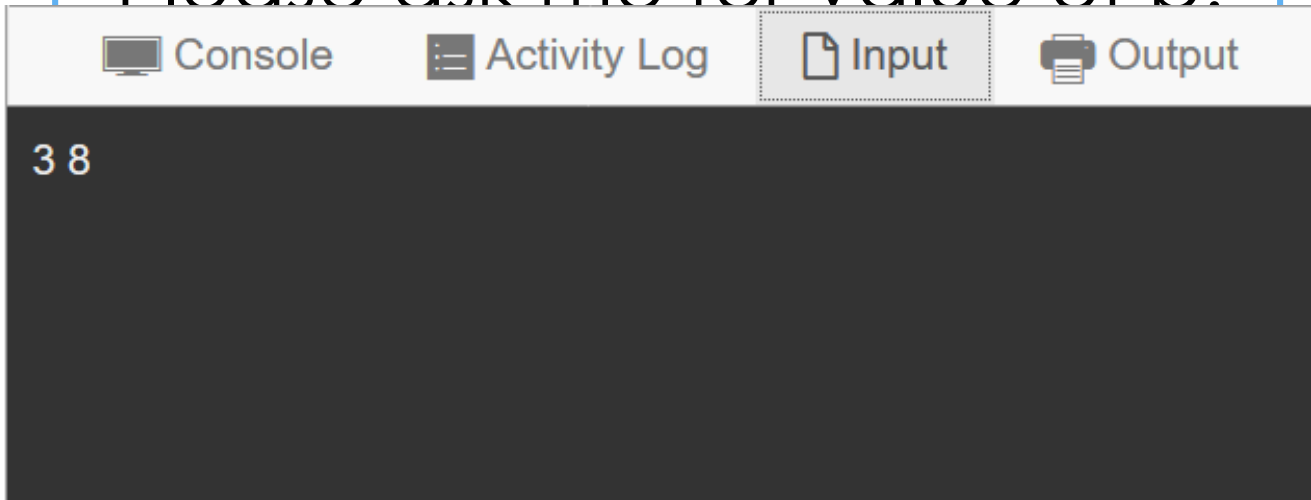
Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

Please ask me for value of b.

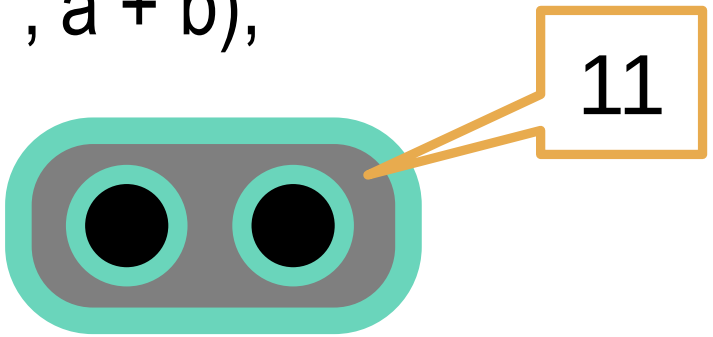


Adding two unknown numbers

HOW WE MUST SPEAK TO MR. COMPILER

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

3 a 8 b 11



HOW WE USUALLY SPEAK TO A HUMAN

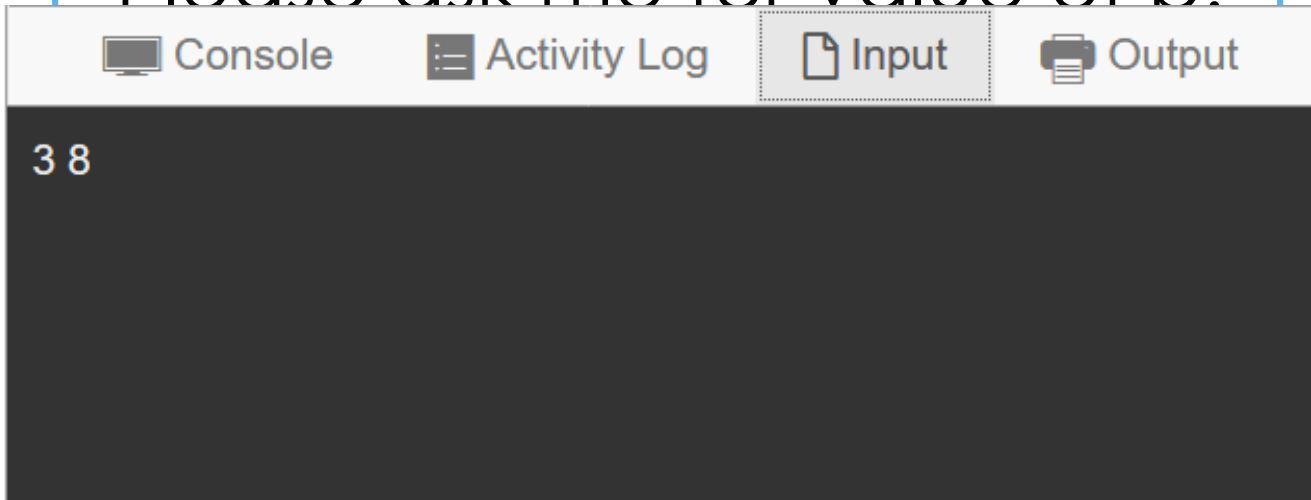
Do you speak English?

Hello

a, b are two variables.

Please ask me for value of a.

Please ask me for value of b.



Words of Caution

Words of Caution

In Prutor, input has to be specified before “Execute”

Words of Caution

In Prutor, input has to be specified before “Execute”

Please be very careful about this common mistake

Words of Caution

In Prutor, input has to be specified before “Execute”

Please be very careful about this common mistake

```
scanf(“%d”,a);
```

Words of Caution

In Prutor, input has to be specified before “Execute”

Please be very careful about this common mistake

scanf(“%d”,a); 

Words of Caution

In Prutor, input has to be specified before “Execute”

Please be very careful about this common mistake

scanf(“%d”,a);  scanf(“%d”,&a);

Words of Caution

In Prutor, input has to be specified before “Execute”

Please be very careful about this common mistake

scanf(“%d”,a); 

scanf(“%d”,&a); 

Words of Caution

In Prutor, input has to be specified before “Execute”

Please be very careful about this common mistake

scanf(“%d”,a); 

scanf(“%d”,&a); 

Will explain what this & means, in a few weeks

Words of Caution

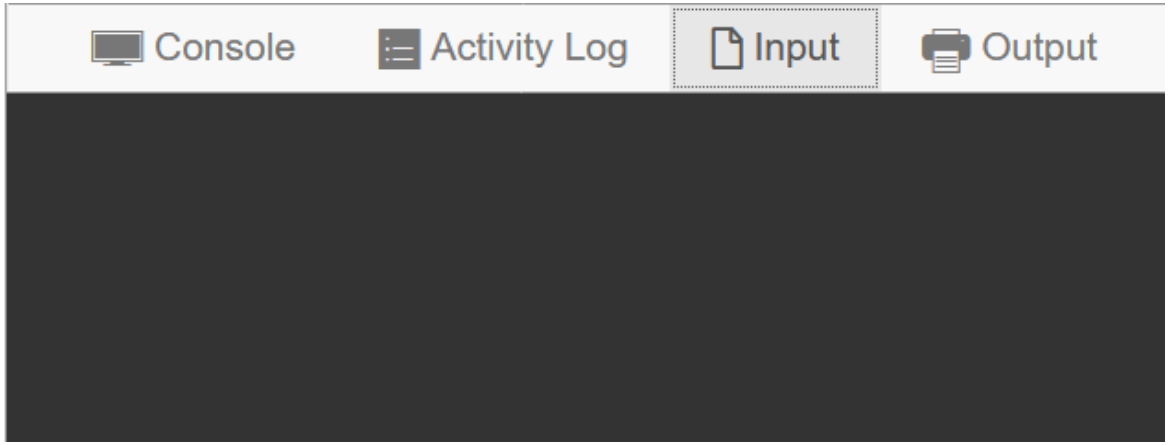
In Prutor, input has to be specified before “Execute”

Please be very careful about this common mistake

scanf(“%d”,a); 

scanf(“%d”,&a); 

Will explain what this & means, in a few weeks



Words of Caution

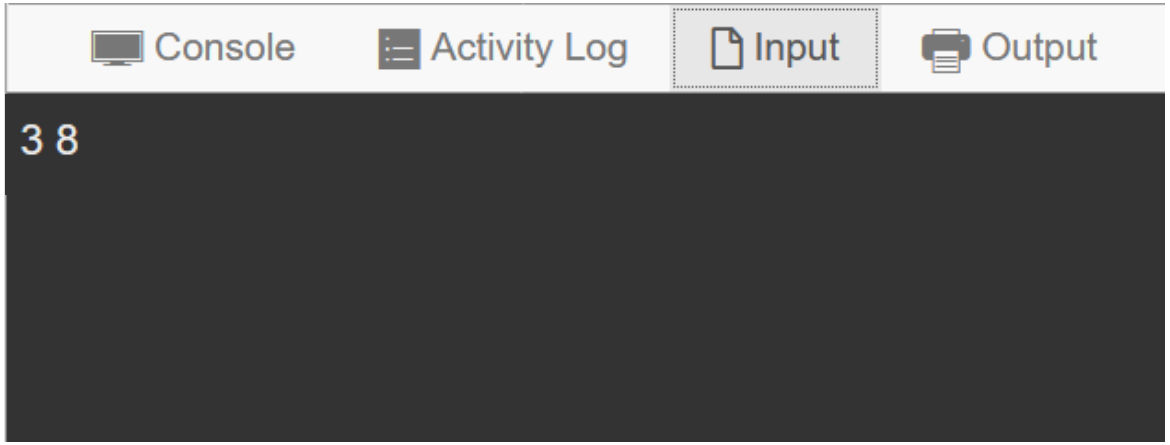
In Prutor, input has to be specified before “Execute”

Please be very careful about this common mistake

scanf(“%d”,a); 

scanf(“%d”,&a); 

Will explain what this & means, in a few weeks



Words of Caution

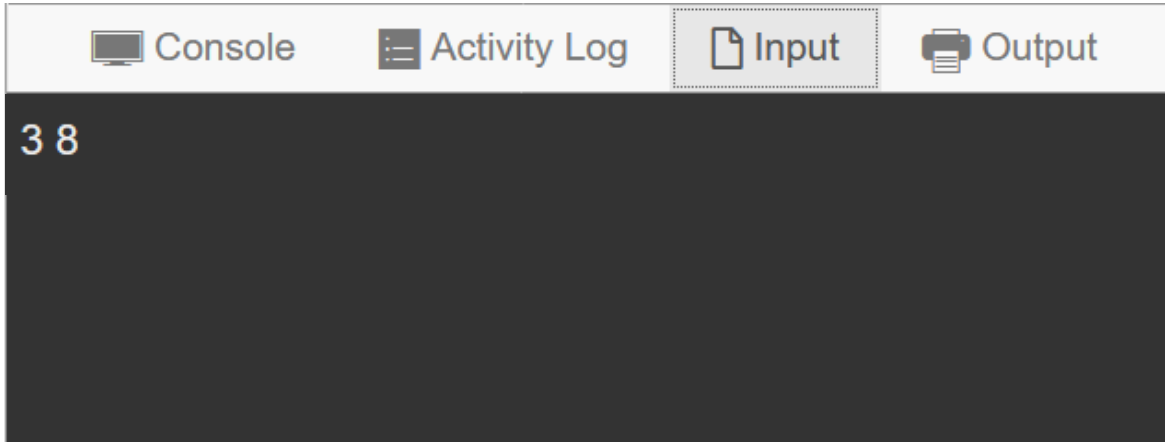
In Prutor, input has to be specified before “Execute”

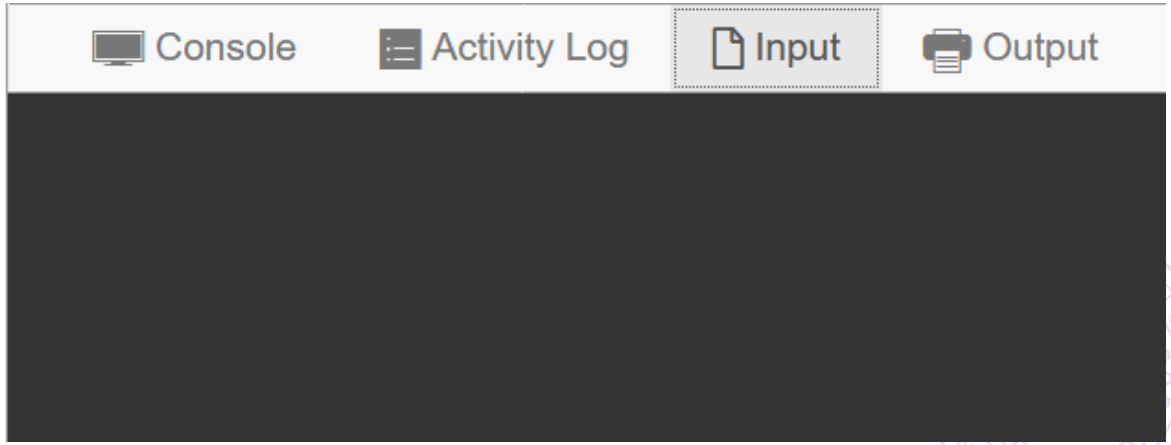
Please be very careful about this common mistake

scanf(“%d”,a); 

scanf(“%d”,&a); 

Will explain what this & means, in a few weeks





Words of Caution

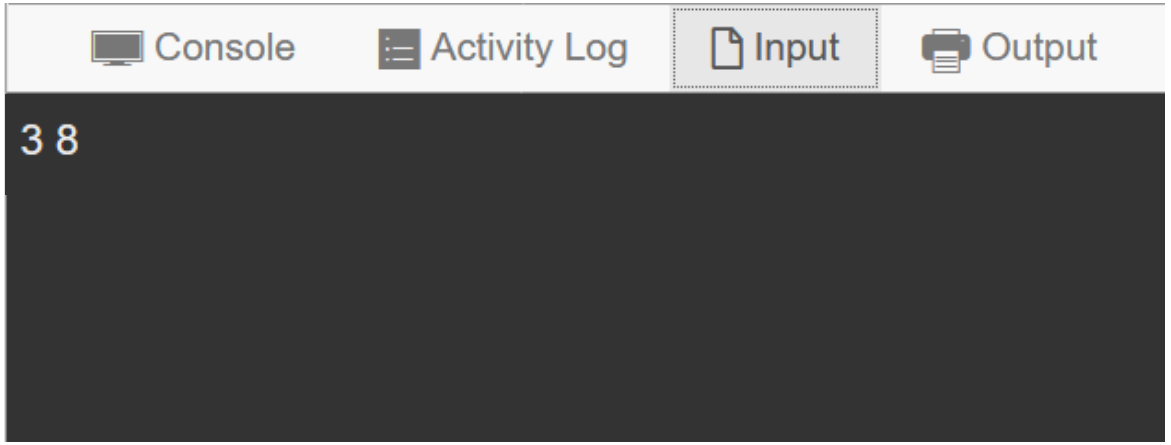
In Prutor, input has to be specified before “Execute”

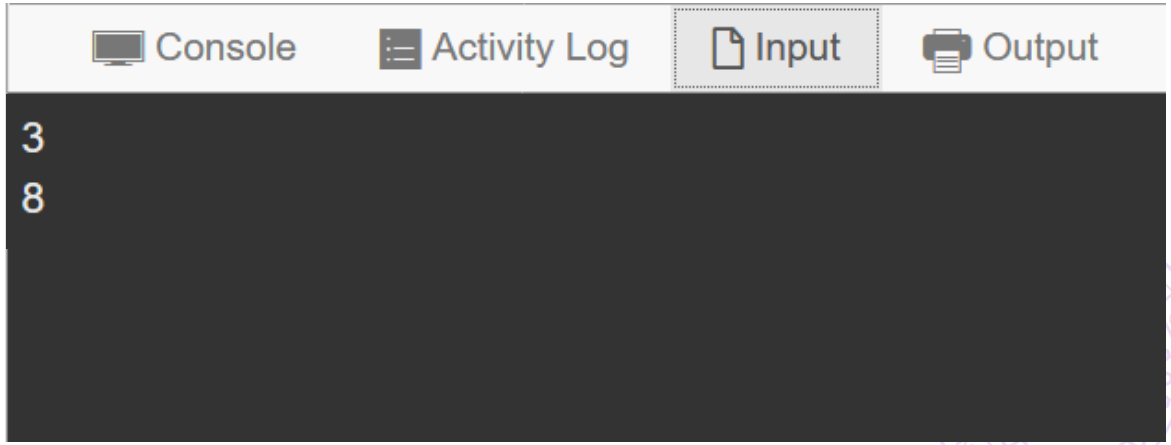
Please be very careful about this common mistake

scanf(“%d”,a); 

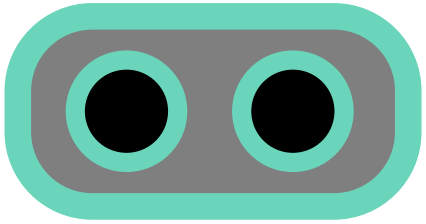
scanf(“%d”,&a); 

Will explain what this & means, in a few weeks





Words of Caution



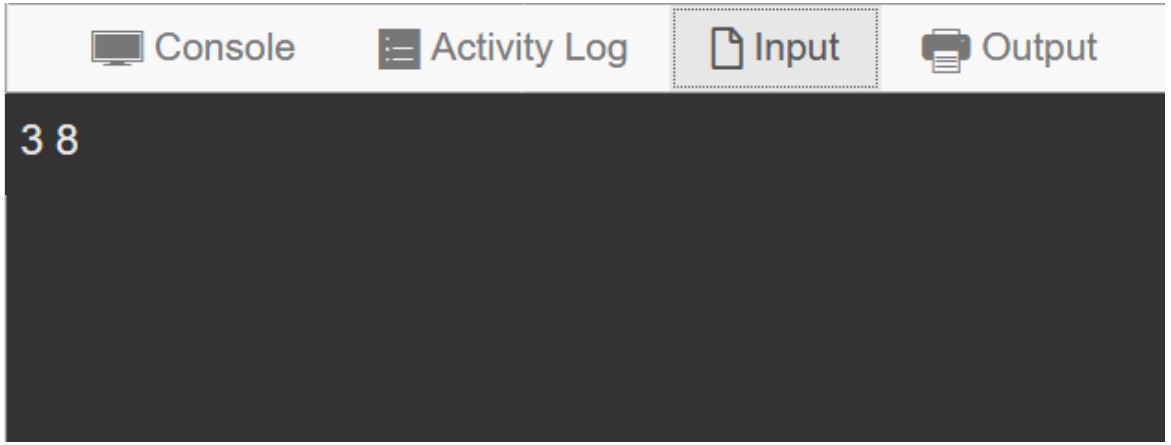
In Prutor, input has to be specified before “Execute”

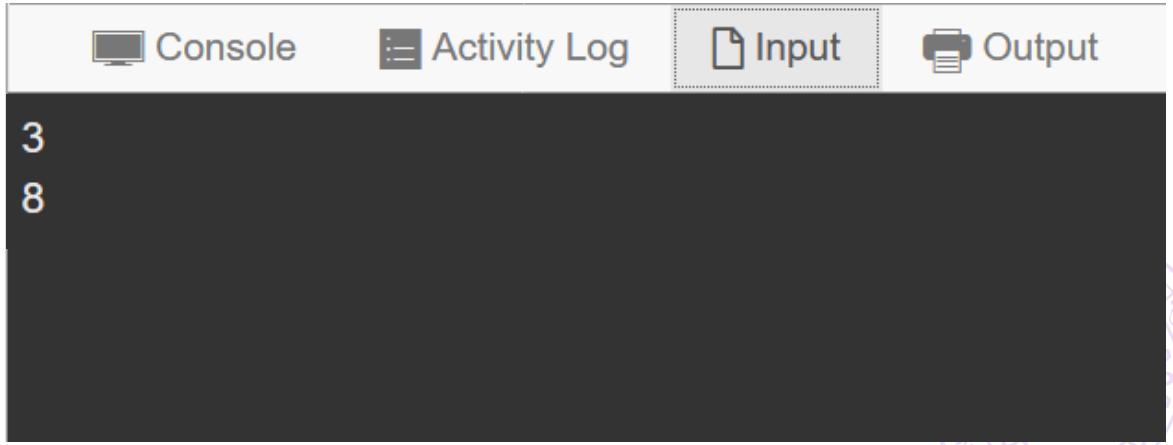
Please be very careful about this common mistake

scanf(“%d”,a); 

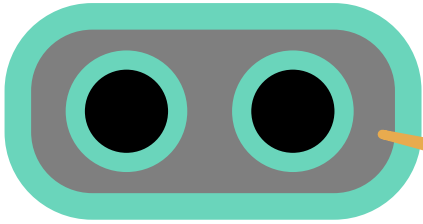
scanf(“%d”,&a); 

Will explain what this & means, in a few weeks





Words of Caution



Both work!
Experiment!

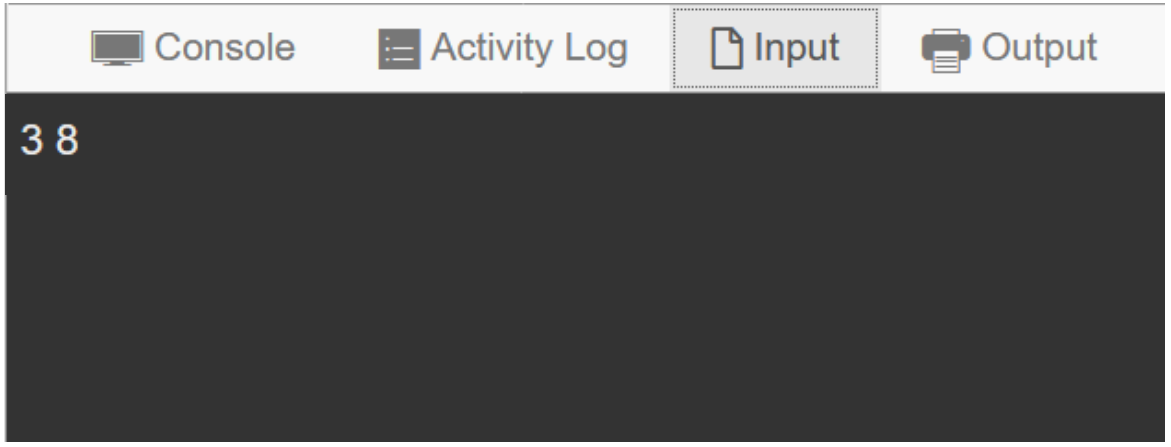
In Prutor, input has to be specified before "E

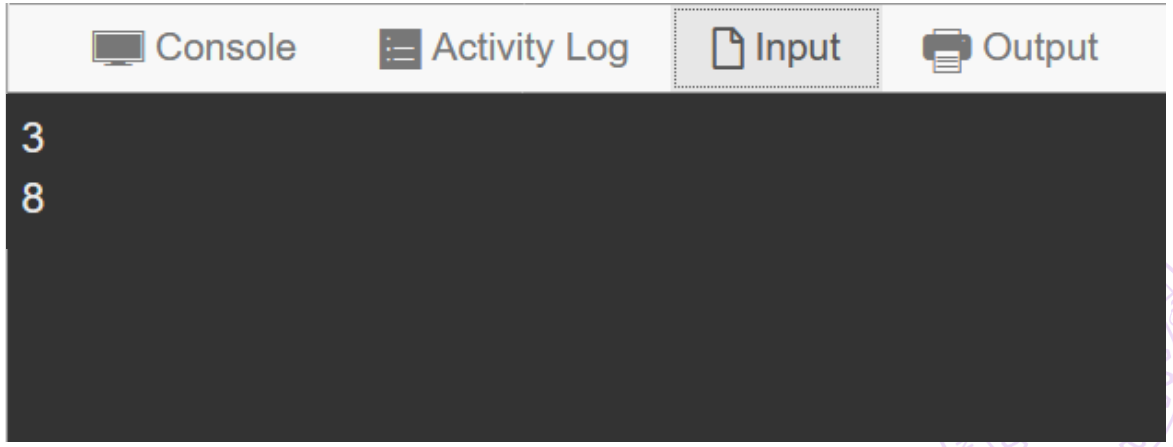
Please be very careful about this common m

scanf("%d",a);

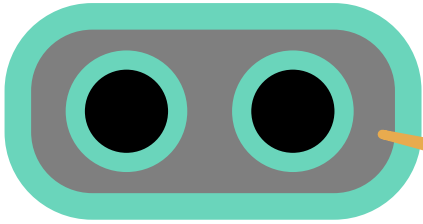
scanf("%d",&a);

Will explain what this & means, in a few weeks





Words of Caution



In Prutor, input has to be specified before "E

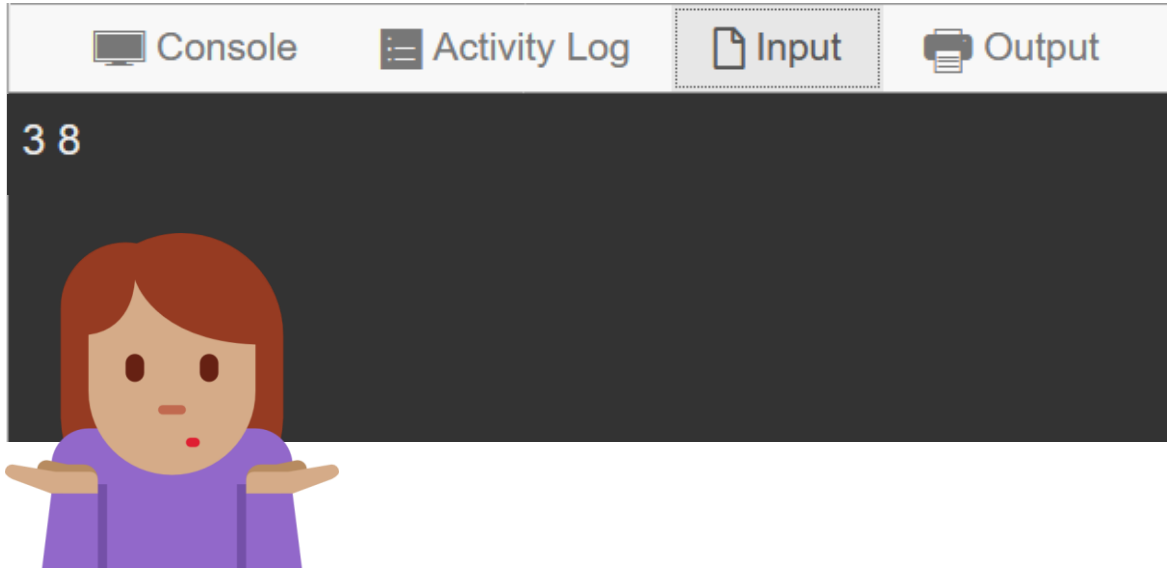
Please be very careful about this common m

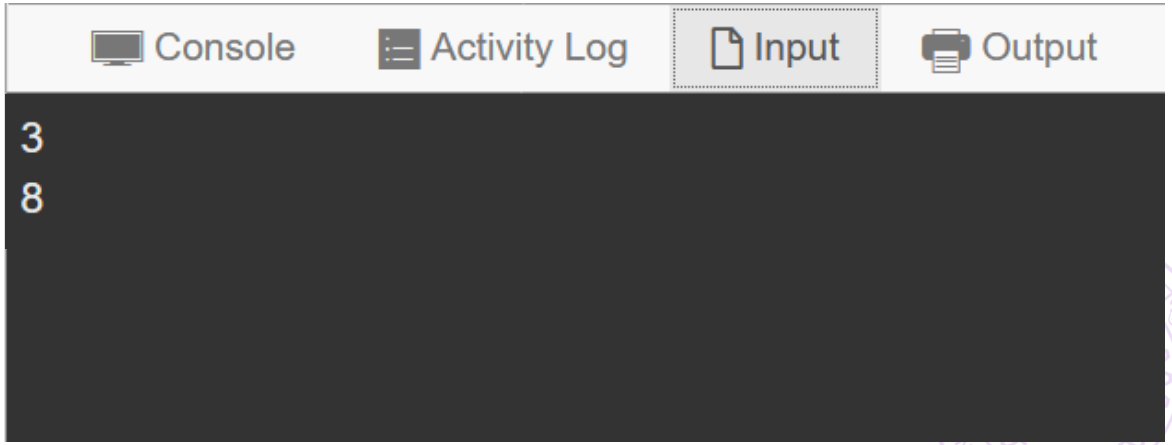
Both work!
Experiment!

scanf("%d",a);

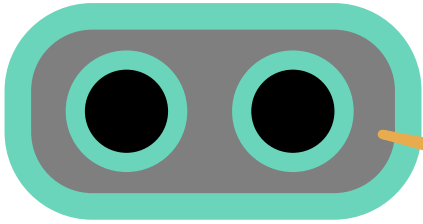
scanf("%d",&a);

Will explain what this & means, in a few weeks





Words of Caution



In Prutor, input has to be specified before "E

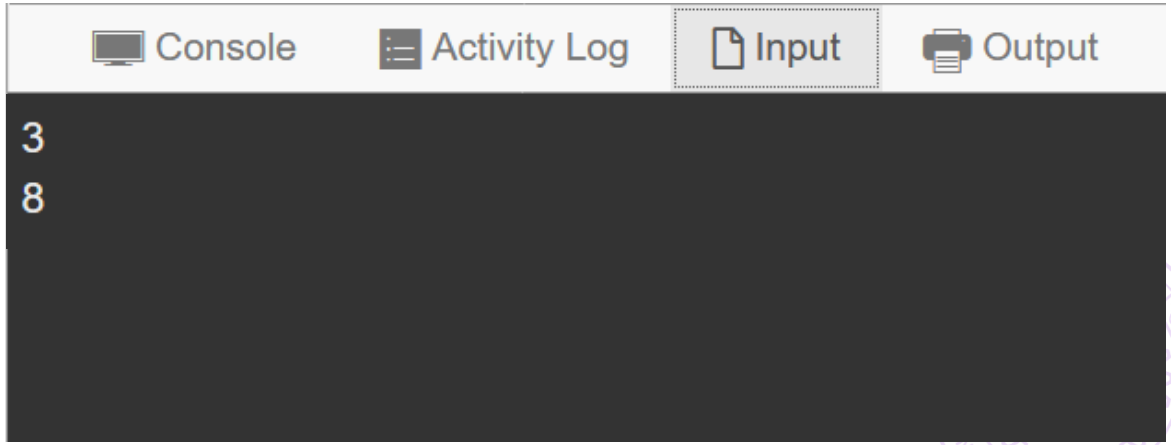
Please be very careful about this common m

Both work!
Experiment!

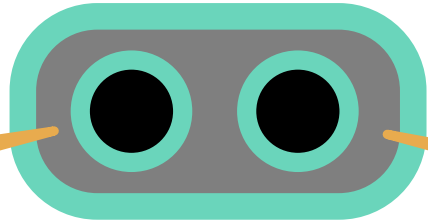
scanf("%d",a);

scanf("%d",&a);

Will explain what this & means, in a few weeks



Words of Caution



Yes, in printf they are different but in scanf, both look like *whitespaces* to me

Both work! Experiment!

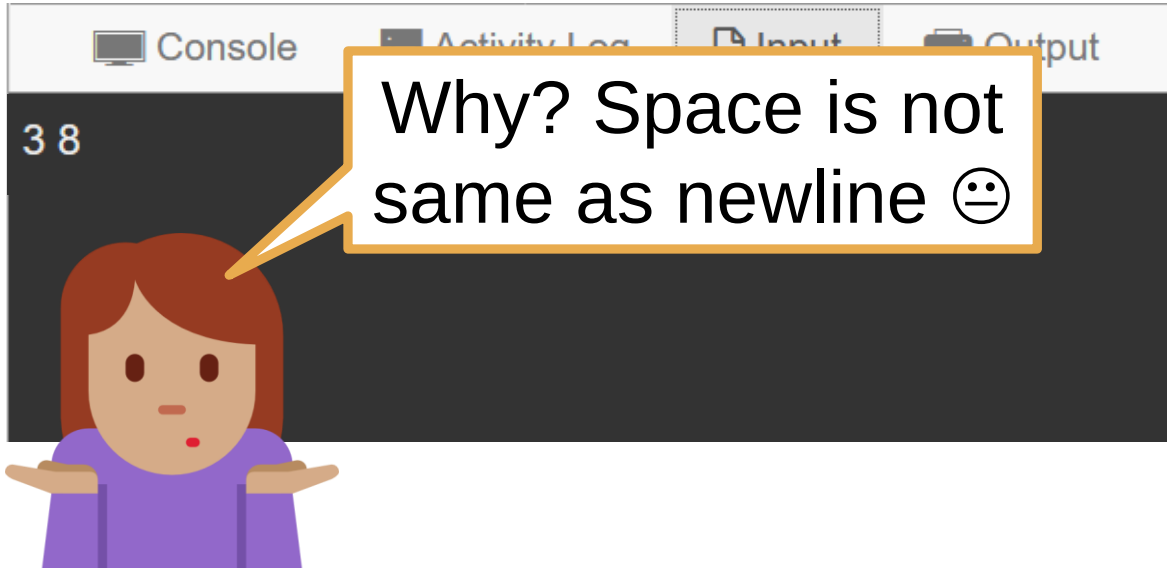
ecified before "E

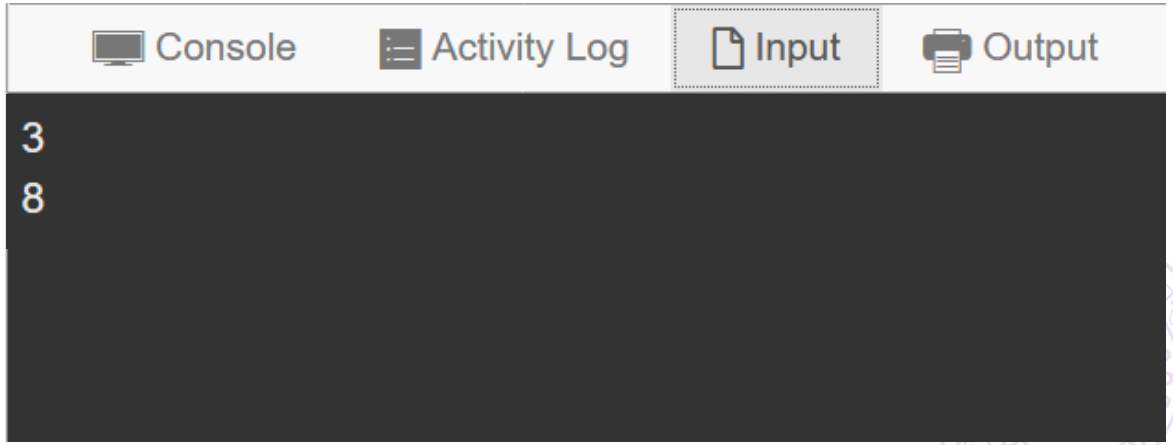
Please be very careful about this common mistake

scanf("%d",a);

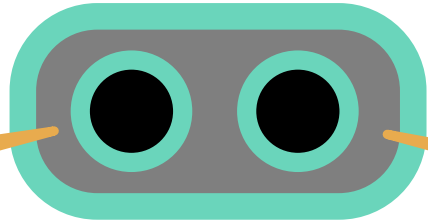
scanf("%d",&a);

Will explain what this & means, in a few weeks





Words of Caution



Yes, in printf they are different but in scanf, both look like *whitespaces* to me

ecified before "E

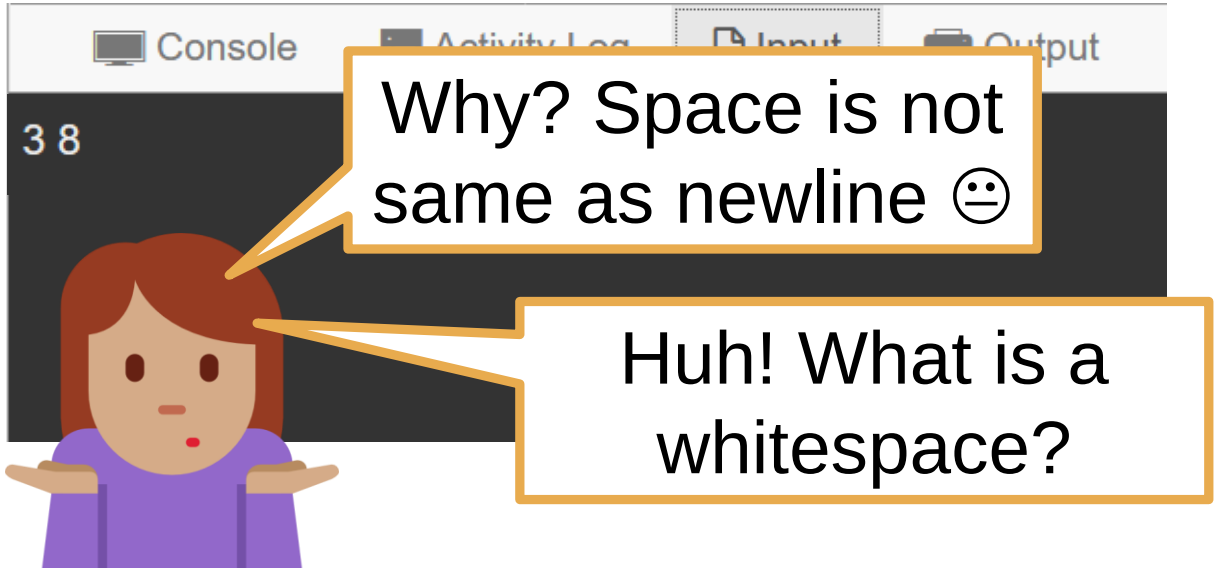
Both work! Experiment!

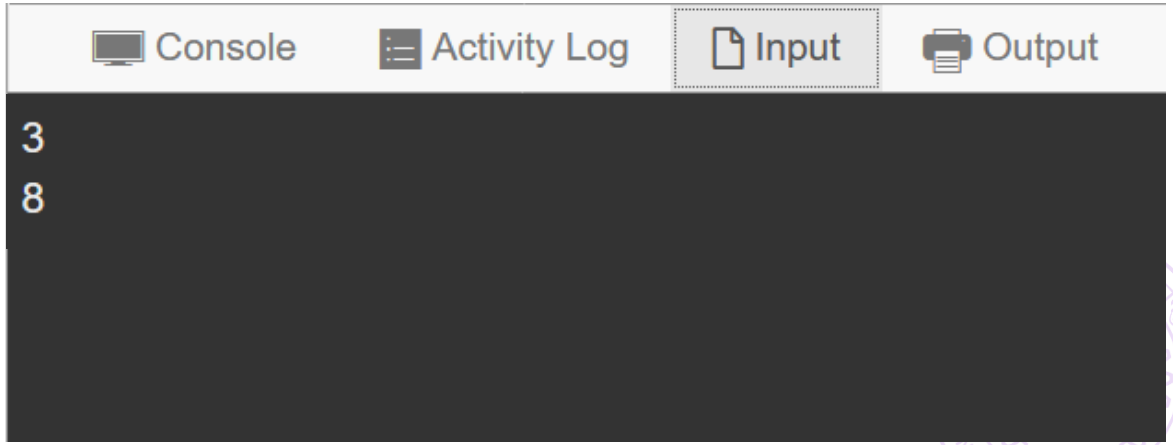
Please be very careful about this common mistake

scanf("%d",a);

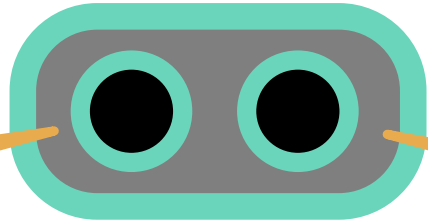
scanf("%d",&a);

Will explain what this & means, in a few weeks





Words of Caution



Yes, in printf they are different but in scanf, both look like *whitespaces* to me

Both work! Experiment!

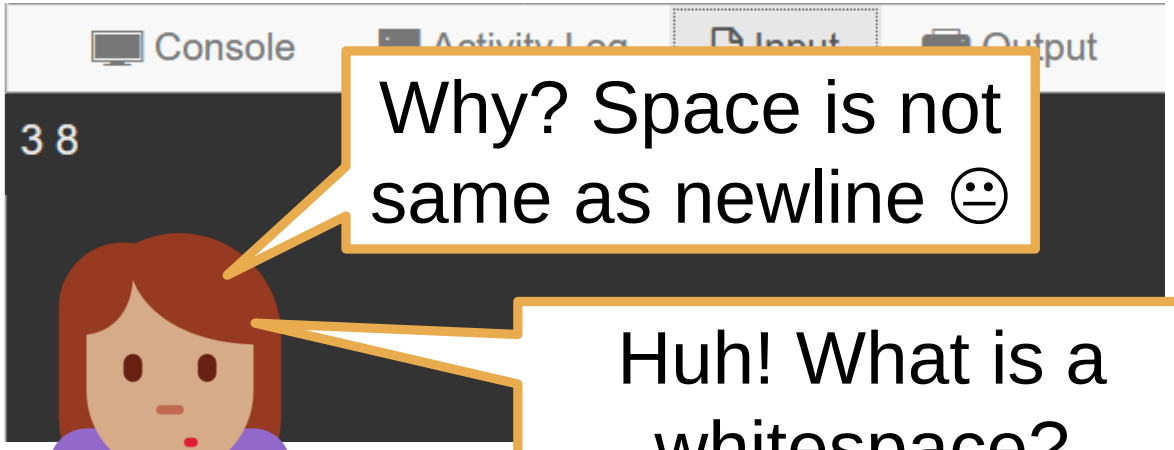
ecified before "E

ut this common

anf("%d",&a);

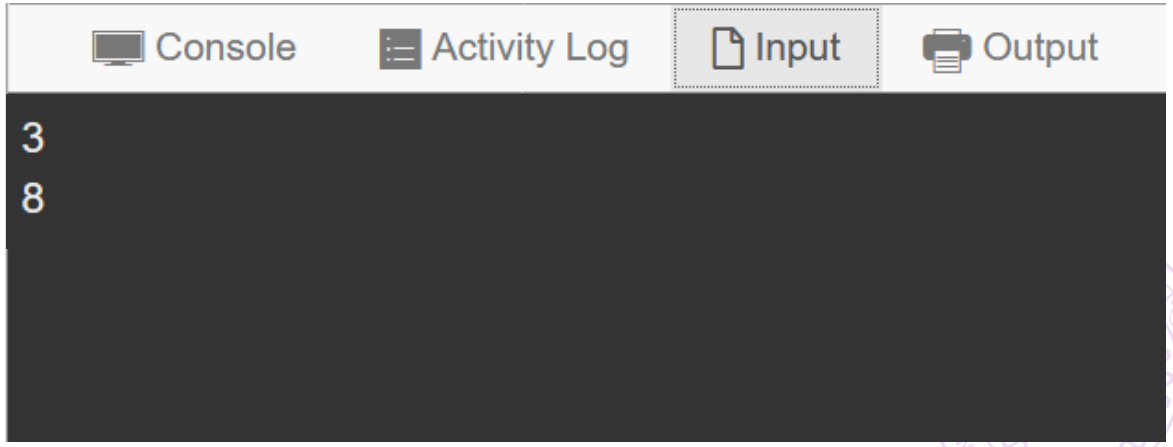
Space, Tab, Newline are called whitespace characters since they are invisible ☺

Will explain what this & means, in a few weeks

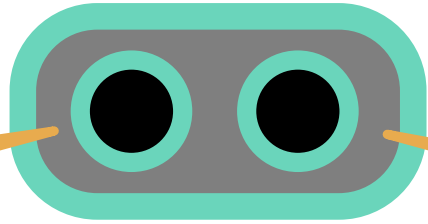


Why? Space is not same as newline 😐

Huh! What is a whitespace?



Words of Caution



Yes, in printf they are different but in scanf, both look like *whitespaces* to me

Both work! Experiment!

ecified before "E

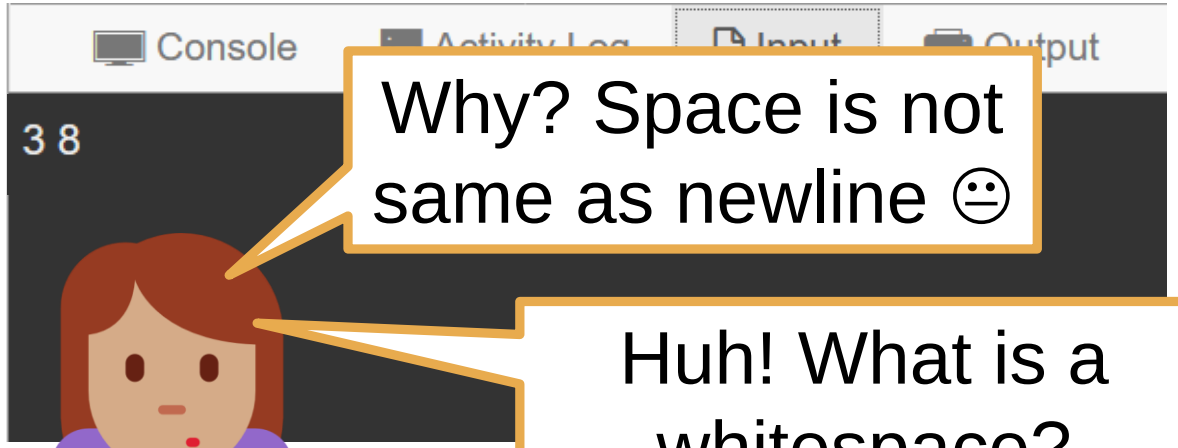
ut this common

anf("%d",&a);

Space, Tab, Newline are called whitespace characters since they are invisible ☺

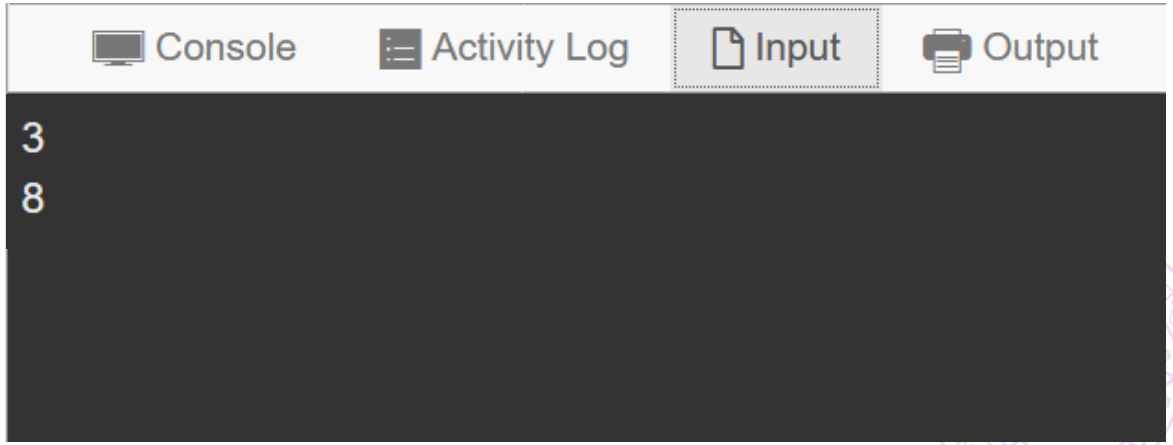
```
3    int a, b, c, d;
```

, in a few weeks

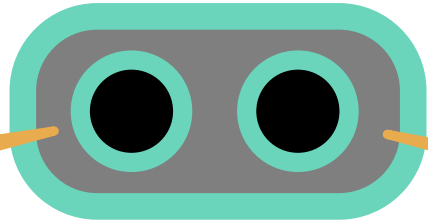


Why? Space is not same as newline 😐

Huh! What is a whitespace?



Words of Caution



Yes, in printf they are different but in scanf, both look like *whitespaces* to me

Both work! Experiment!

ecified before "E

ut this common

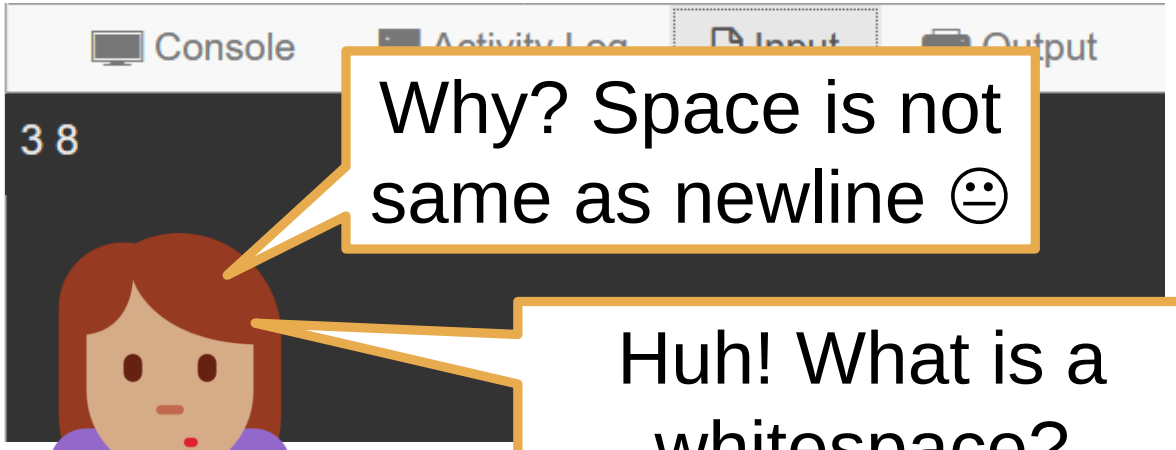
anf("%d",&a);

Space, Tab, Newline are called whitespace characters since they are invisible ☺

```
3    int a, b, c, d;
```

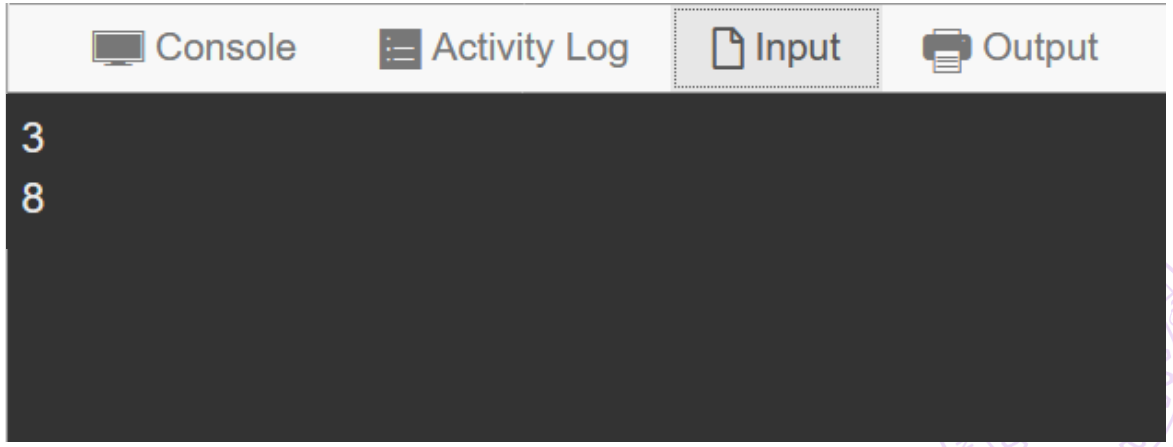
TAB

, in a few weeks

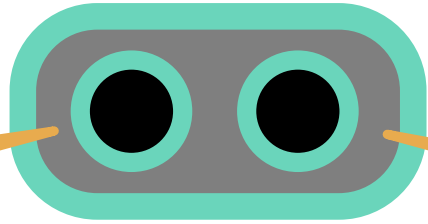


Why? Space is not same as newline 😐

Huh! What is a whitespace?



Words of Caution



Yes, in printf they are different but in scanf, both look like *whitespaces* to me

Both work! Experiment!

ecified before "E

ut this common

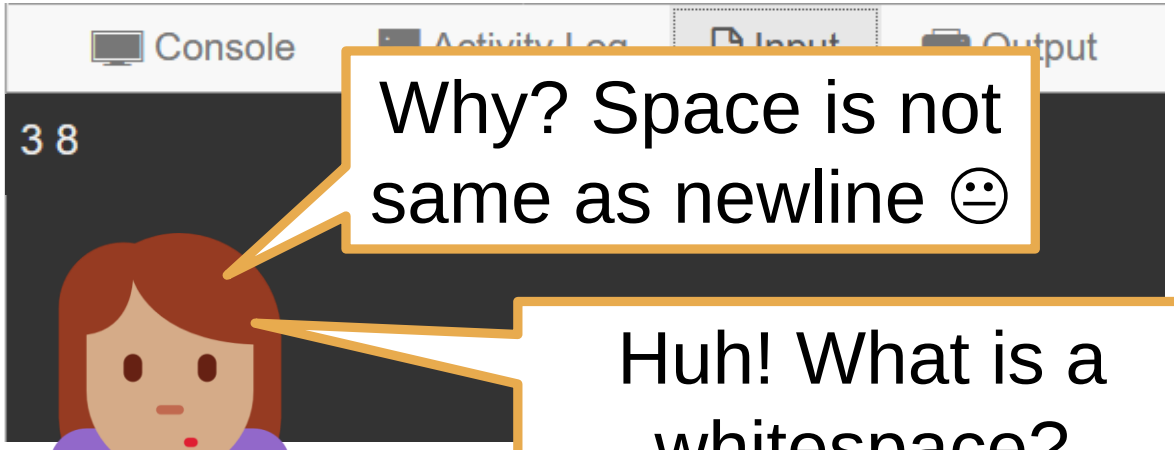
anf("%d",&a);

Space, Tab, Newline are called whitespace characters since they are invisible ☺

```
3    int a,b,c,d;
```

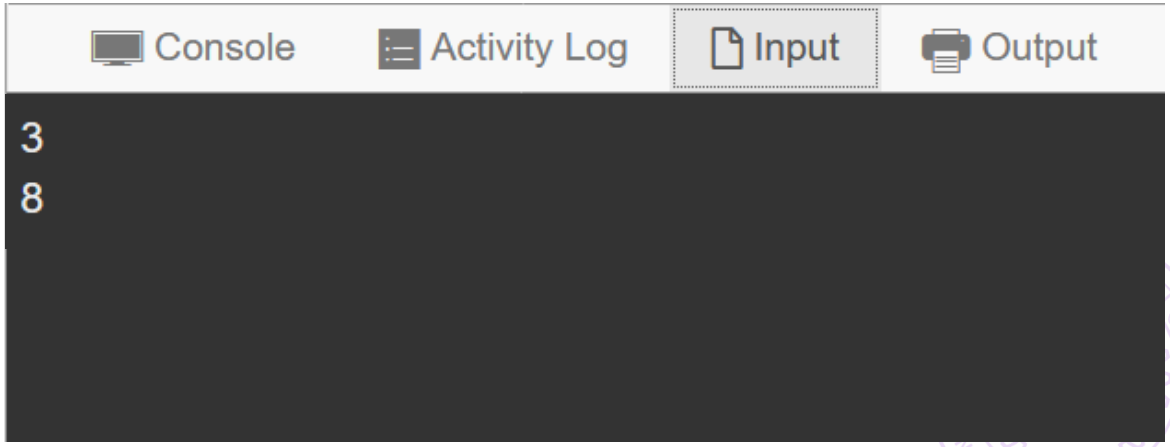
TAB SPACE

Will explain what this & means, in a few weeks

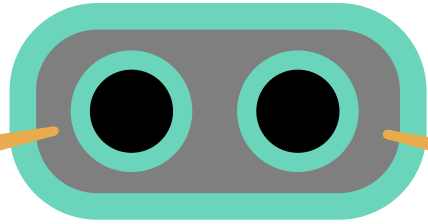


Why? Space is not same as newline 😐

Huh! What is a whitespace?



Words of Caution



Yes, in printf they are different but in scanf, both look like *whitespaces* to me

Both work! Experiment!

ecified before "E

ut this common

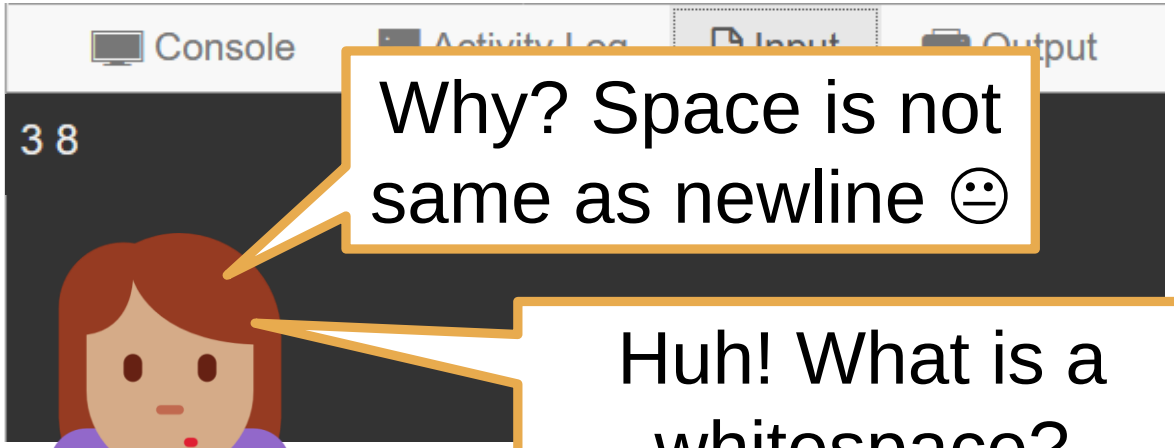
anf("%d",&a);

Space, Tab, Newline are called whitespace characters since they are invisible ☺

```
3    ···int0a,0b,0c,0d;0
```

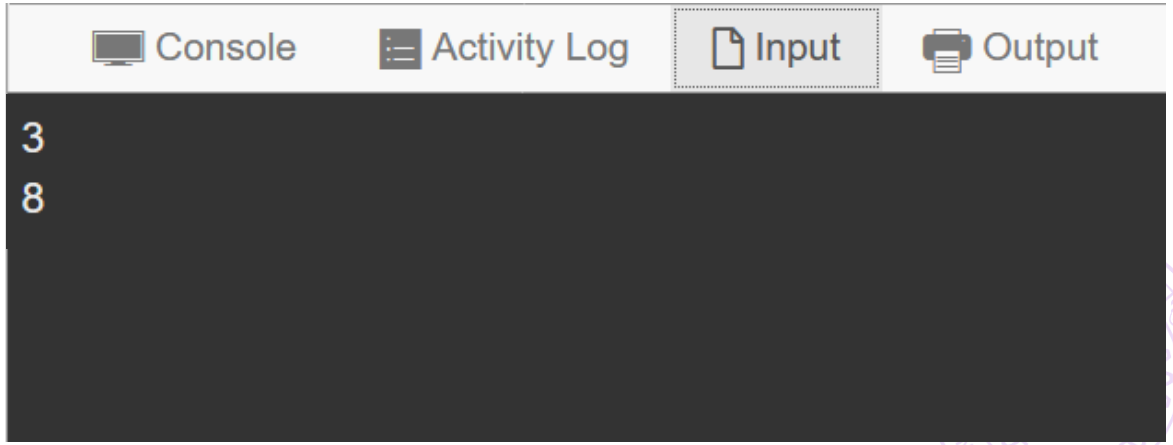
TAB SPACE NEWLINE

in a few weeks



Why? Space is not same as newline 😐

Huh! What is a whitespace?



Shorthand for multiple inputs

Shorthand for multiple inputs

HOW WE USUALLY SPEAK TO A HUMAN

Shorthand for multiple inputs

HOW WE USUALLY SPEAK TO A HUMAN

Shorthand for multiple inputs

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Shorthand for multiple inputs

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

Shorthand for multiple inputs

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Shorthand for multiple inputs

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

Shorthand for multiple inputs

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

Please print their sum.

Shorthand for multiple inputs

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

Please print their sum.

Goodbye

Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

Please print their sum.

Goodbye

Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

Please print their sum.

Goodbye

Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

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

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

Please print their sum.

Goodbye

Shorthand for multiple inputs

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 are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

Please print their sum.

Goodbye

Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

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

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

Please print their sum.

Goodbye

Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
int a, b;
scanf("%d%d", &a, &b);
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

Please print their sum.

Goodbye

Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
int a, b;
scanf("%d%d", &a, &b);
printf("%d", a + b);
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

Please print their sum.

Goodbye

Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

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

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

Please print their sum.

Goodbye

Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

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

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

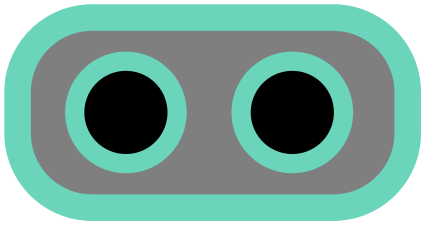
Please print their sum.

Goodbye

Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

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



HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

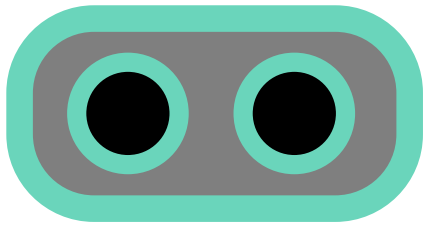
Please print their sum.

Goodbye

Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

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



HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

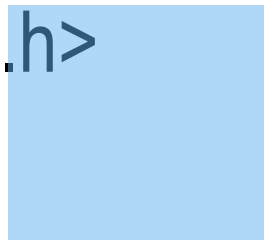
Please print their sum.

Goodbye

Shorthand for multiple inputs

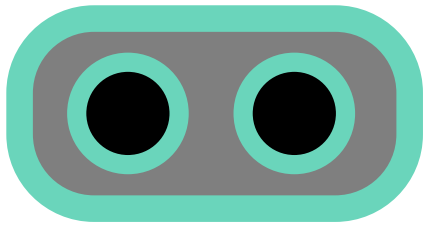
HOW WE MUST SPEAK TO MR. COMPILER

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



a

b



HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

Please print their sum.

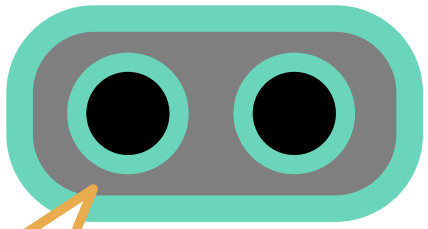
Goodbye

Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

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

a b



Input please

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

Please print their sum.

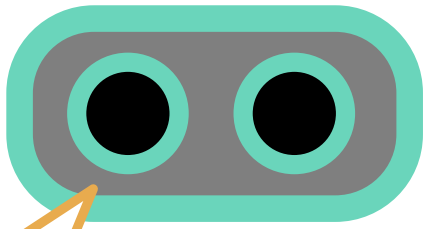
Goodbye

Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

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

a b



Input please

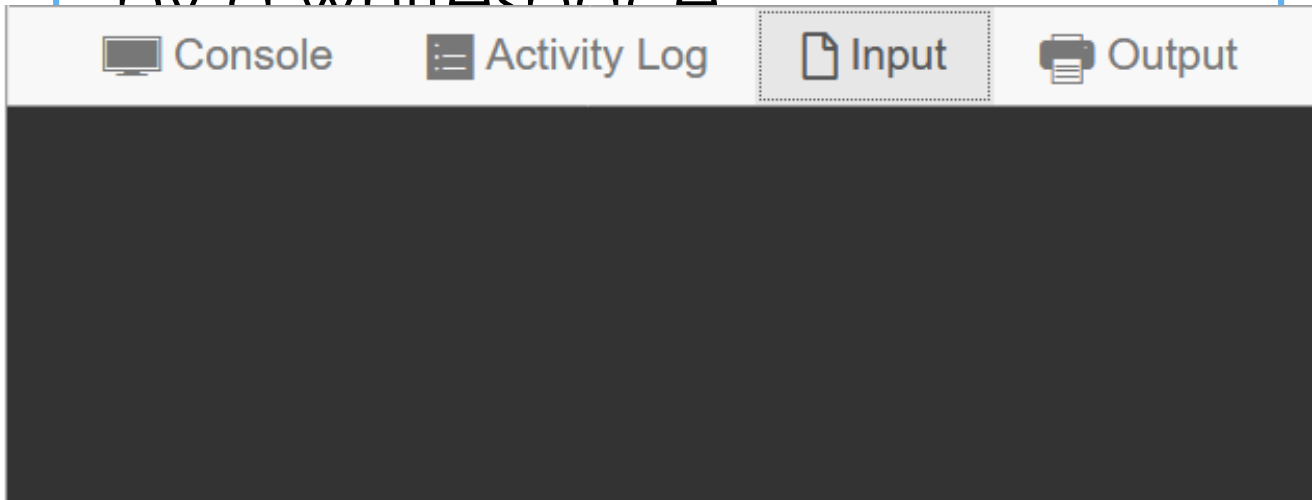
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace



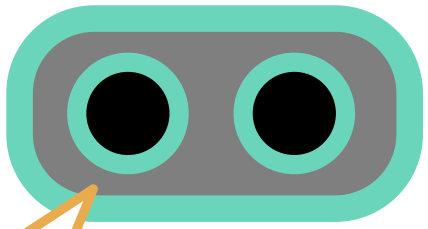
Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

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

a

b



Input please

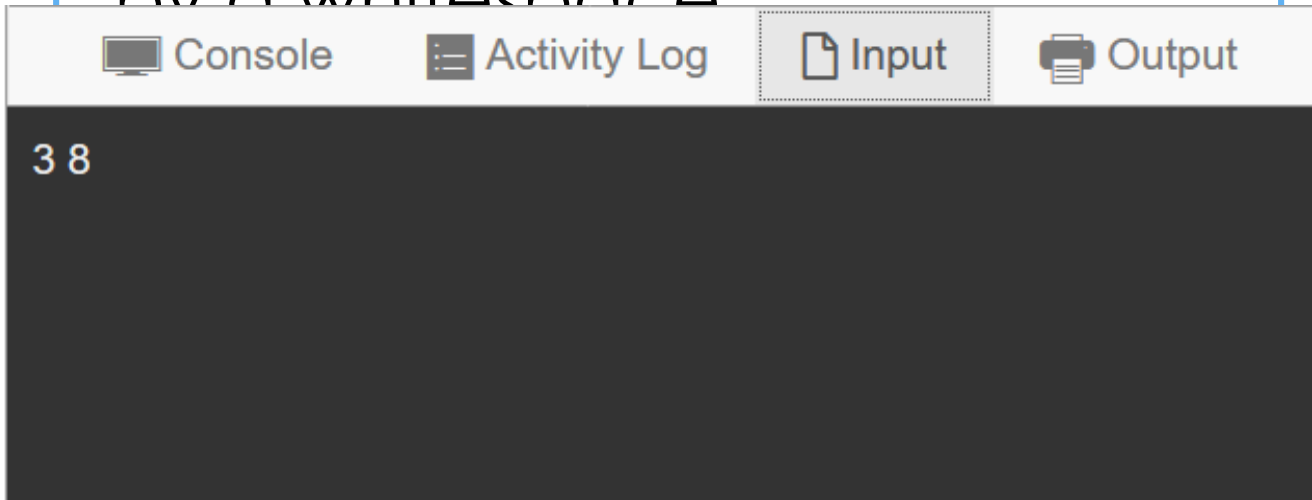
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace



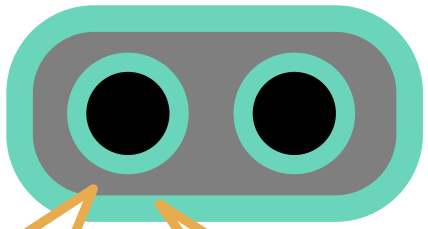
Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

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

a

b



Input please

Thanks

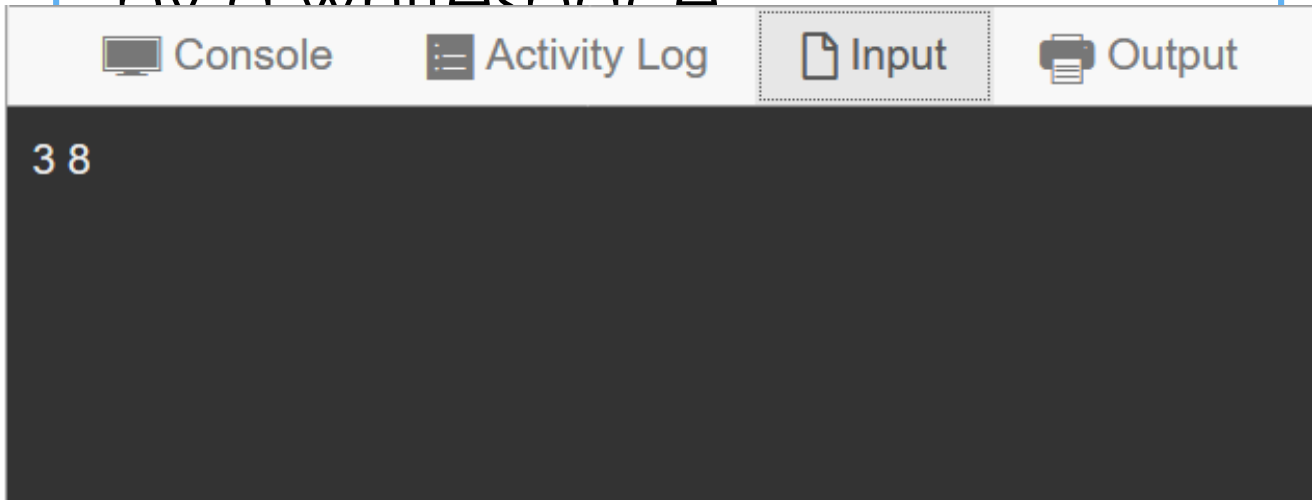
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

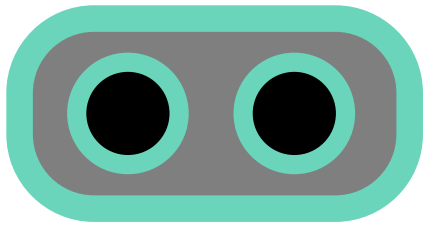


Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

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

a b



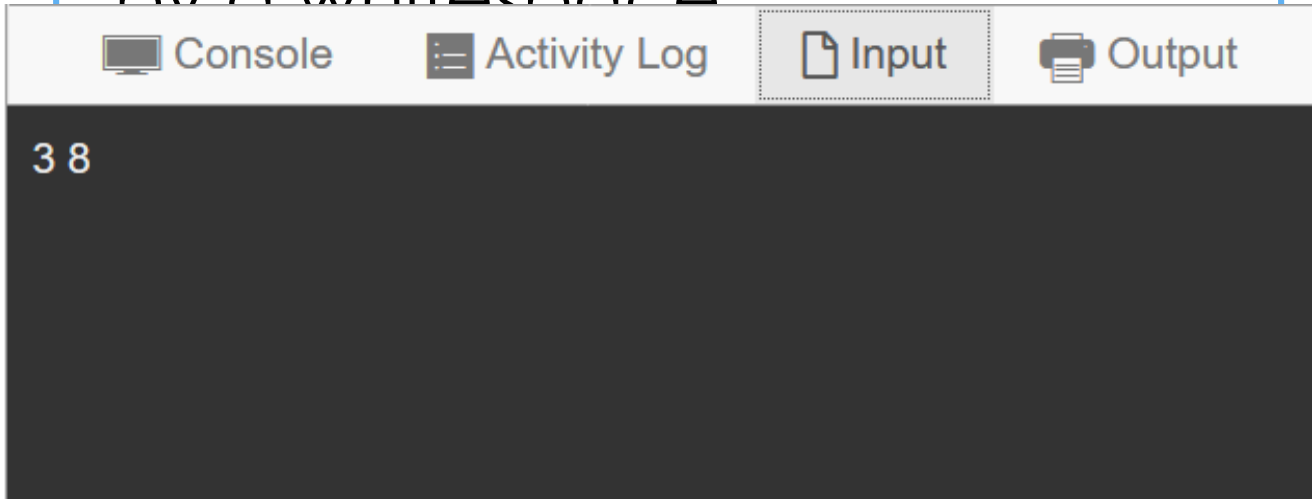
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace



Shorthand for multiple inputs

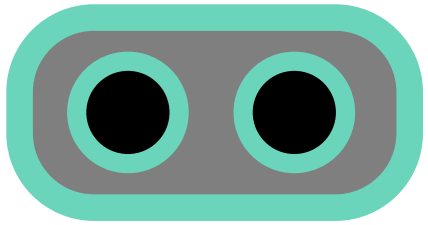
HOW WE MUST SPEAK TO MR. COMPILER

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

3

a

b



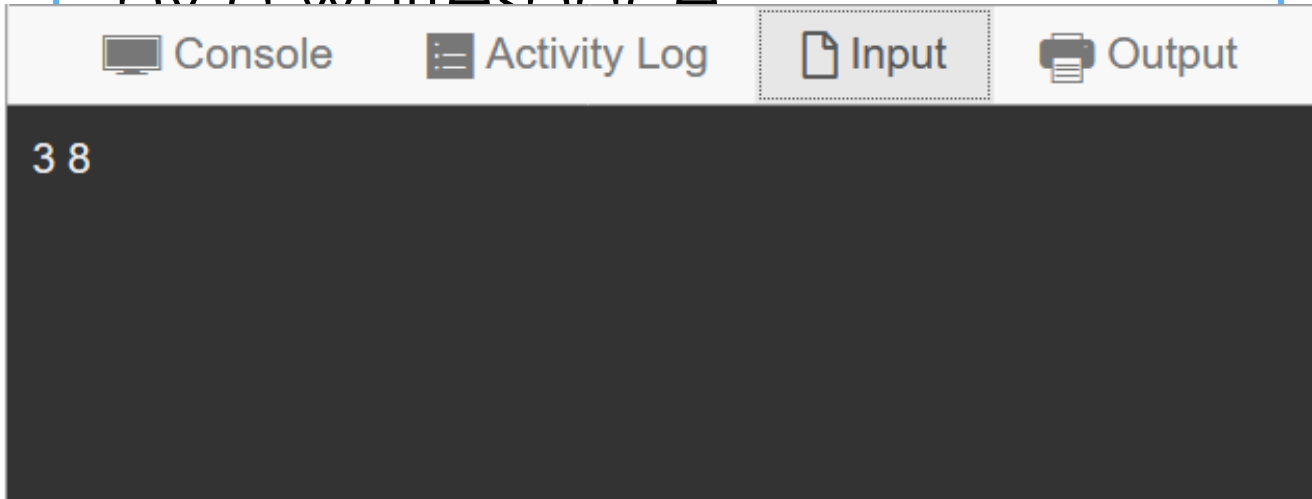
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace



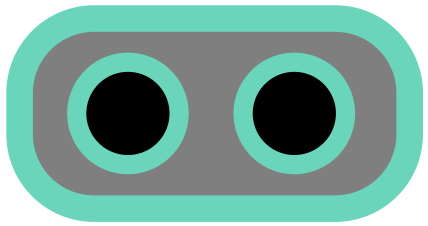
Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

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

3 a

8 b



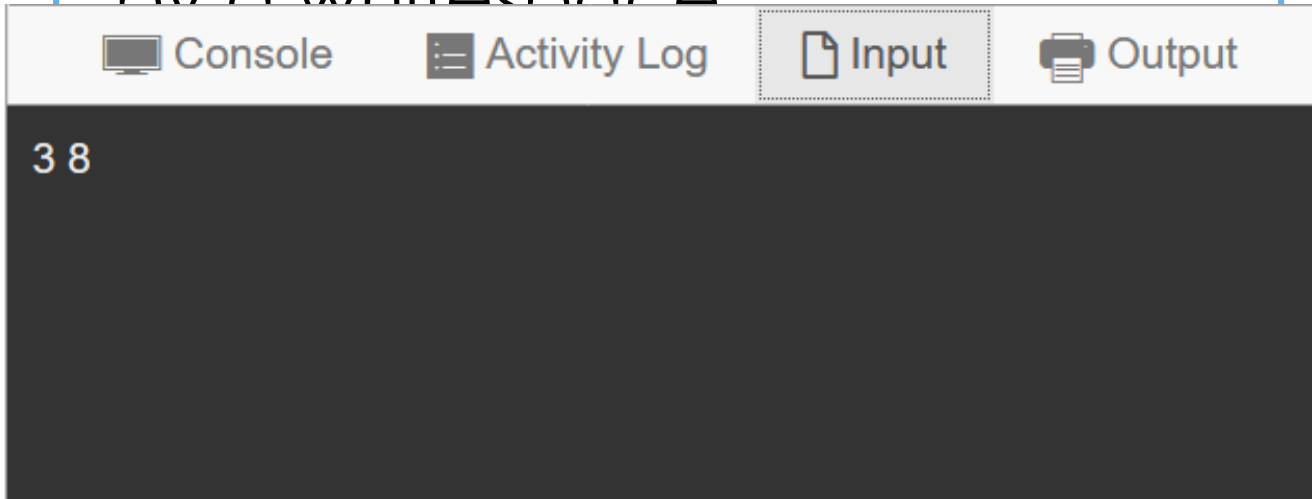
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

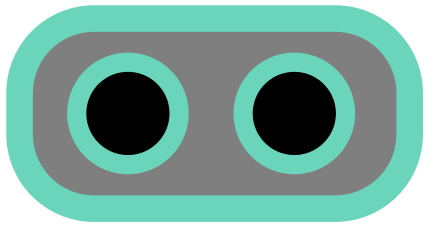


Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

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

3 a 8 b 11



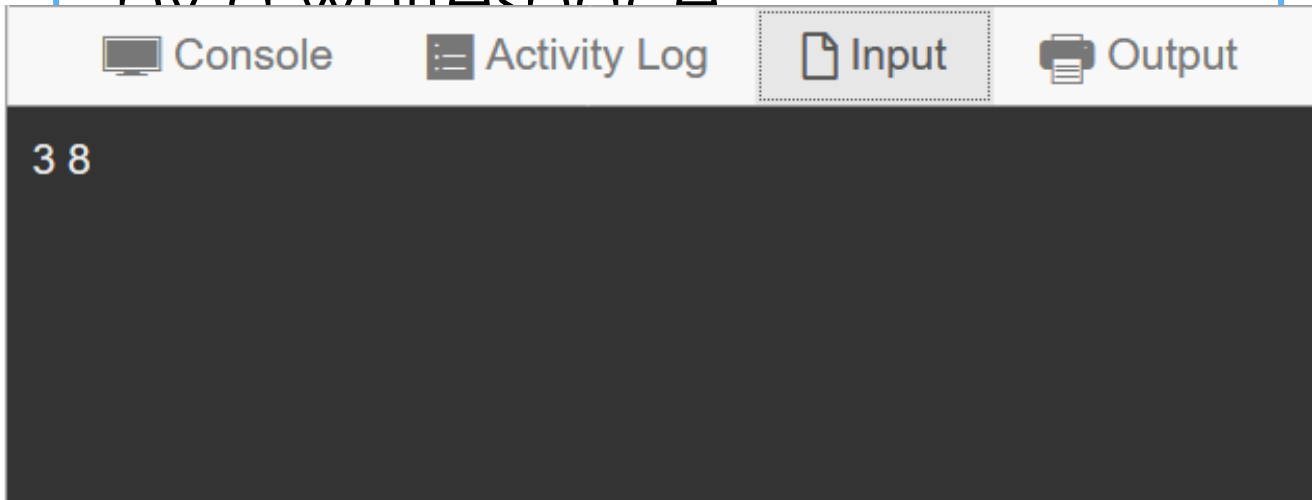
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace



Shorthand for multiple inputs

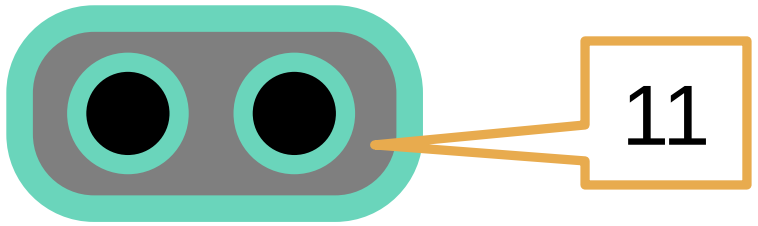
HOW WE MUST SPEAK TO MR. COMPILER

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

3 → a

8 → b

11



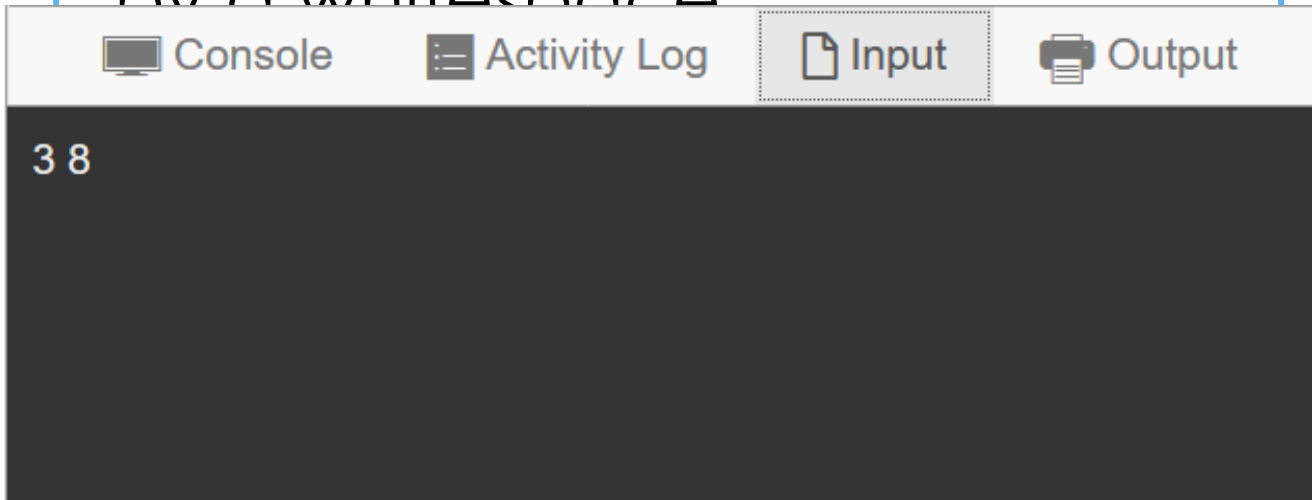
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

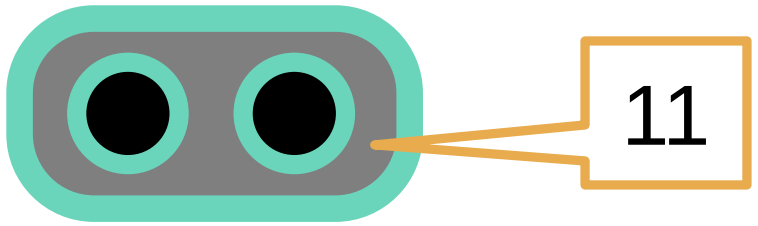


Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

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

3 a 8 b 11



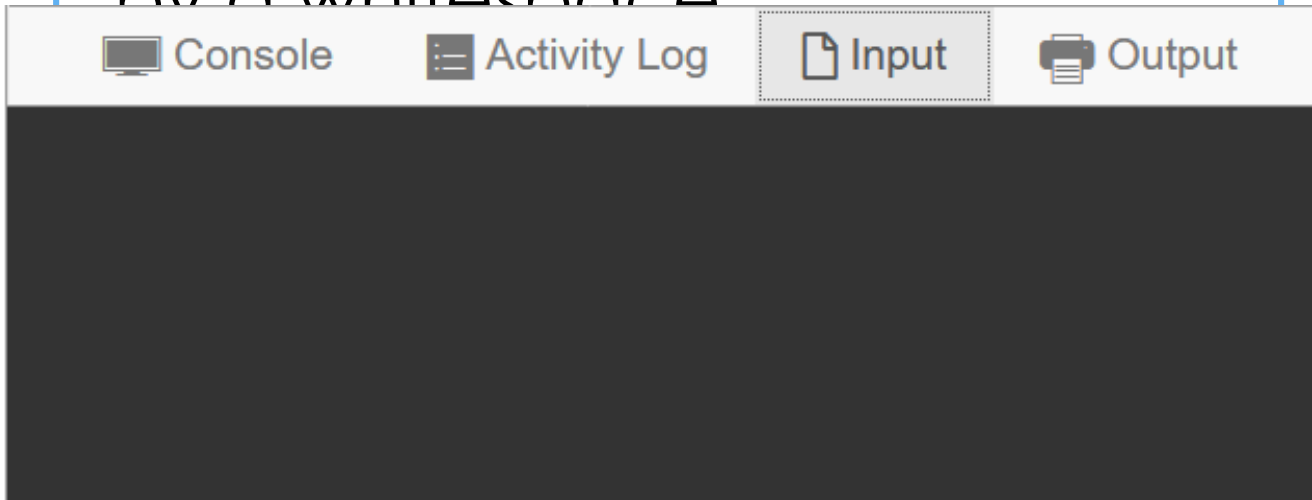
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace



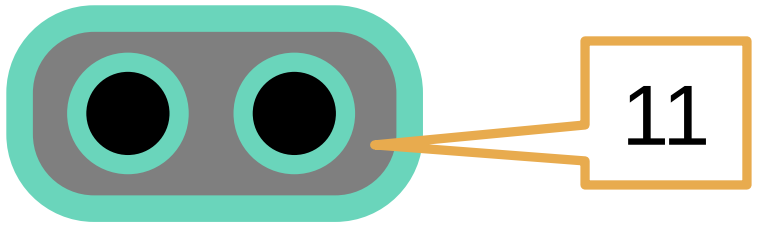
Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

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

3 a 8 b

11



11

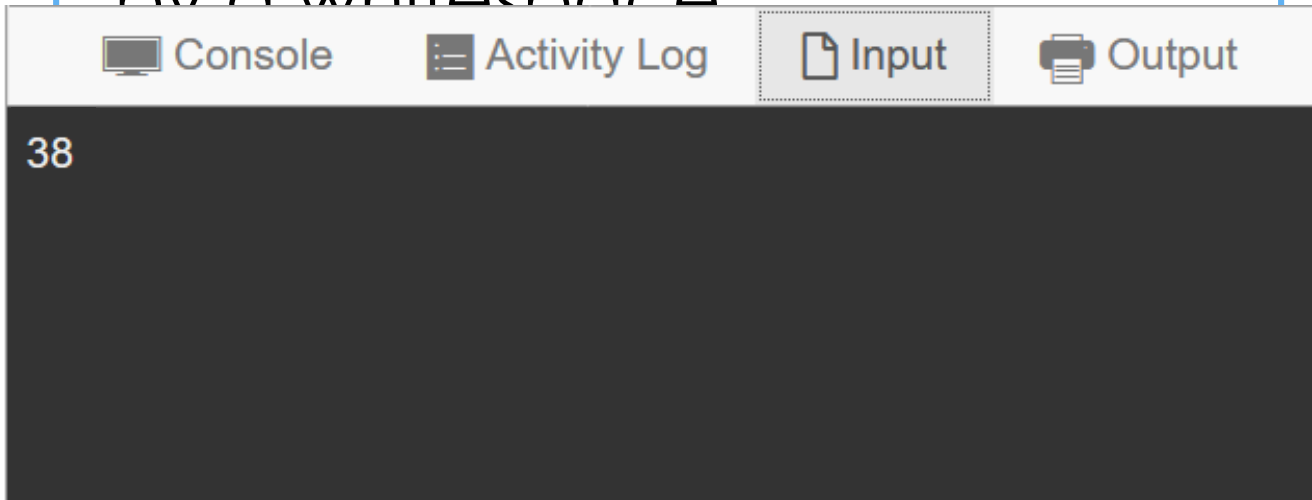
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace



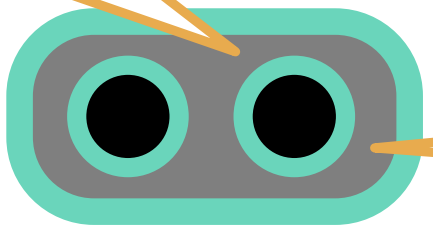
Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
int a, b;
                    &a, &b);
                    b);
return 0;
}
```

3 a 8 b 11

You entered only one integer



11

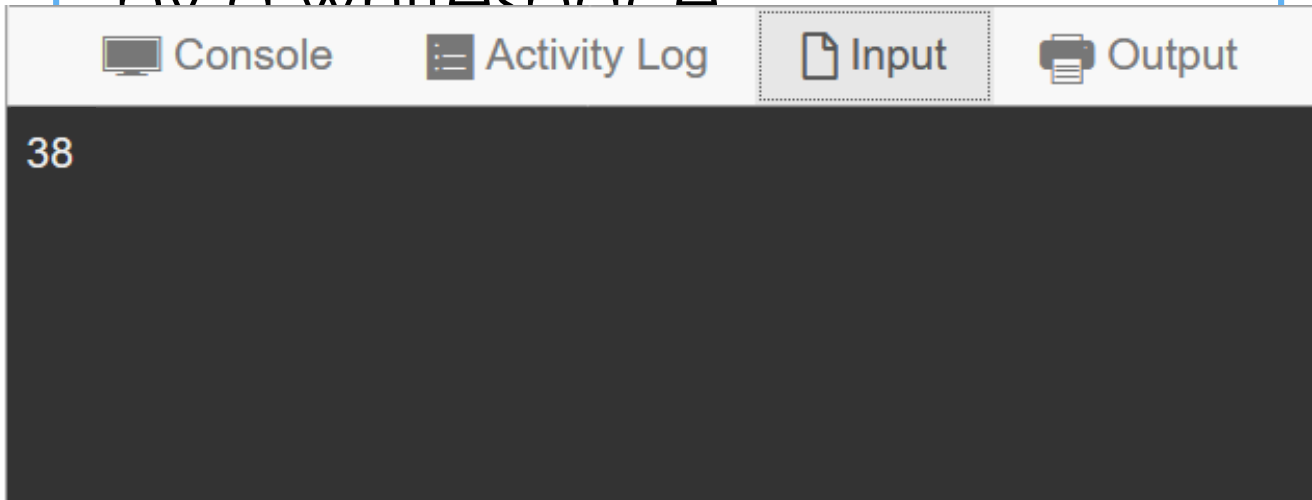
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace



Shorthand for multiple inputs

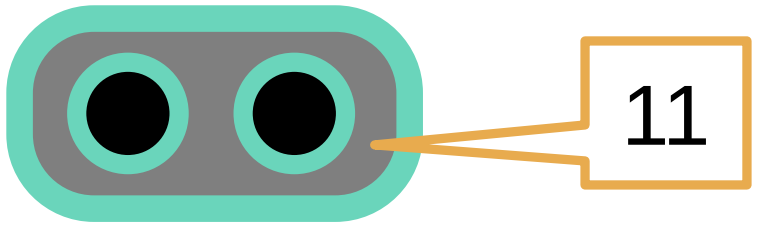
HOW WE MUST SPEAK TO MR. COMPILER

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

3 → a

8 → b

11



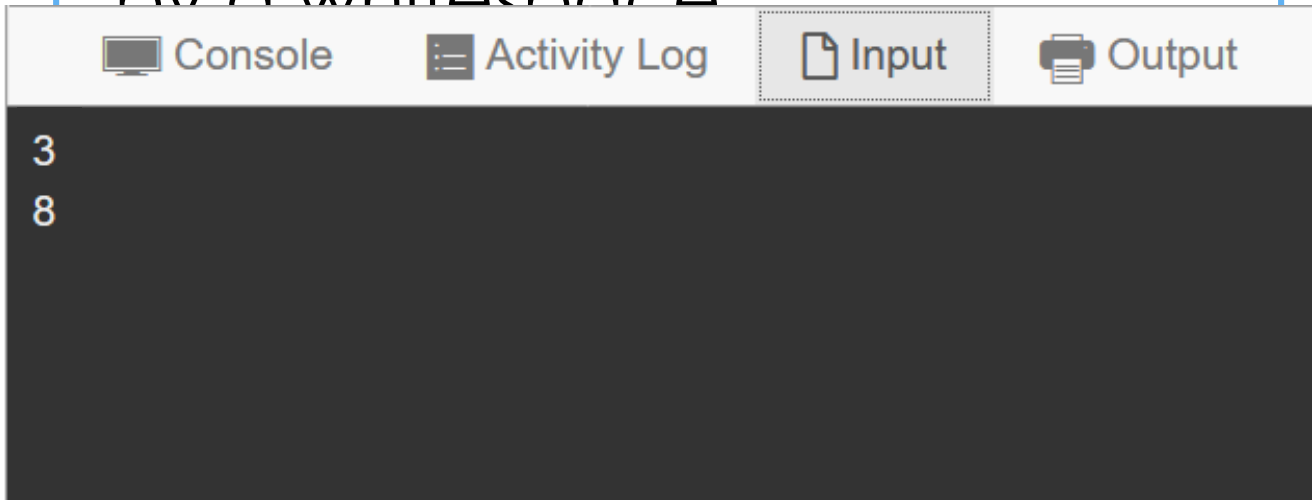
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

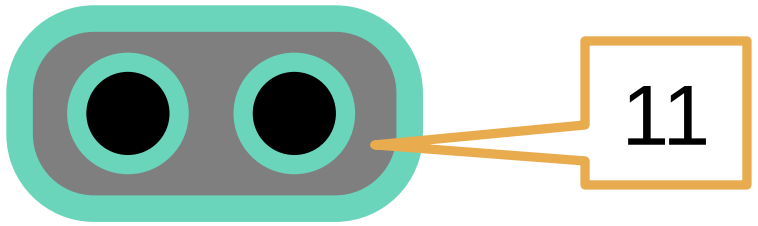


Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

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

3 a 8 b 11



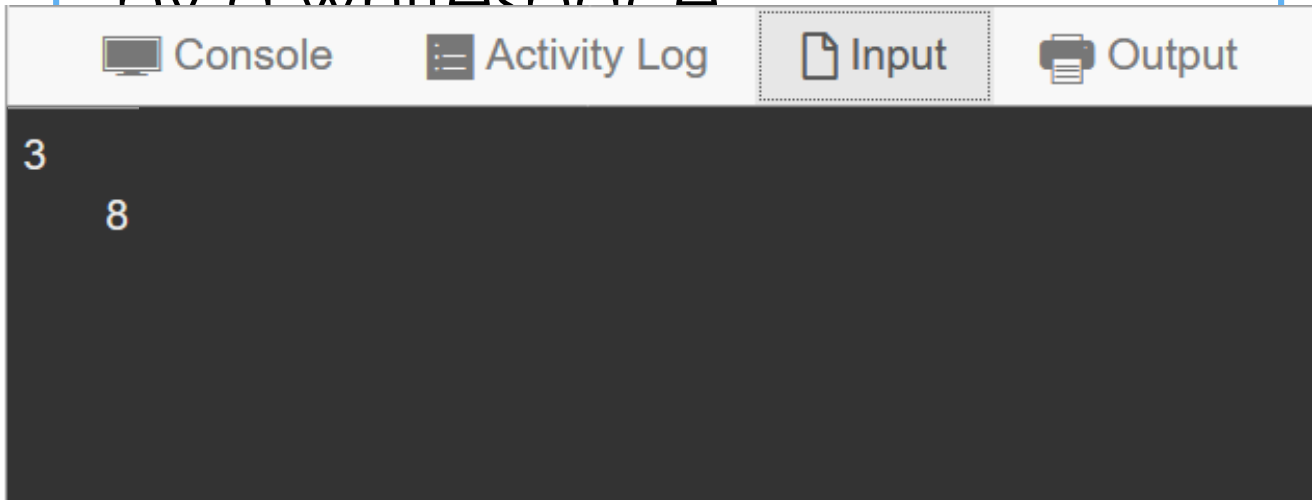
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace



Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
int a, b;
                d", &a, &b);
                + b);
return 0;
}
```

3 8

a b

All look the same to me

11

11

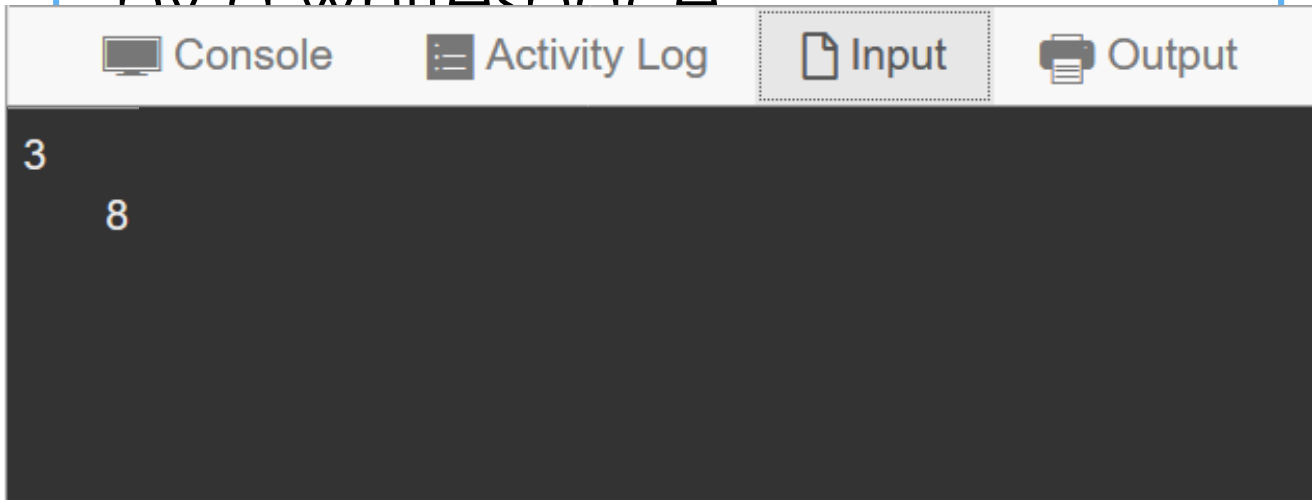
HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace



Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

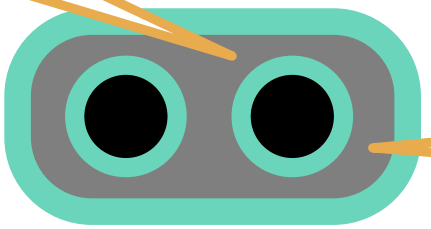
```
#include <stdio.h>
int main(){
int a, b;
                d", &a, &b);
                + b);
return 0;
}
```

3 8

a b

11

All look the same to me



11

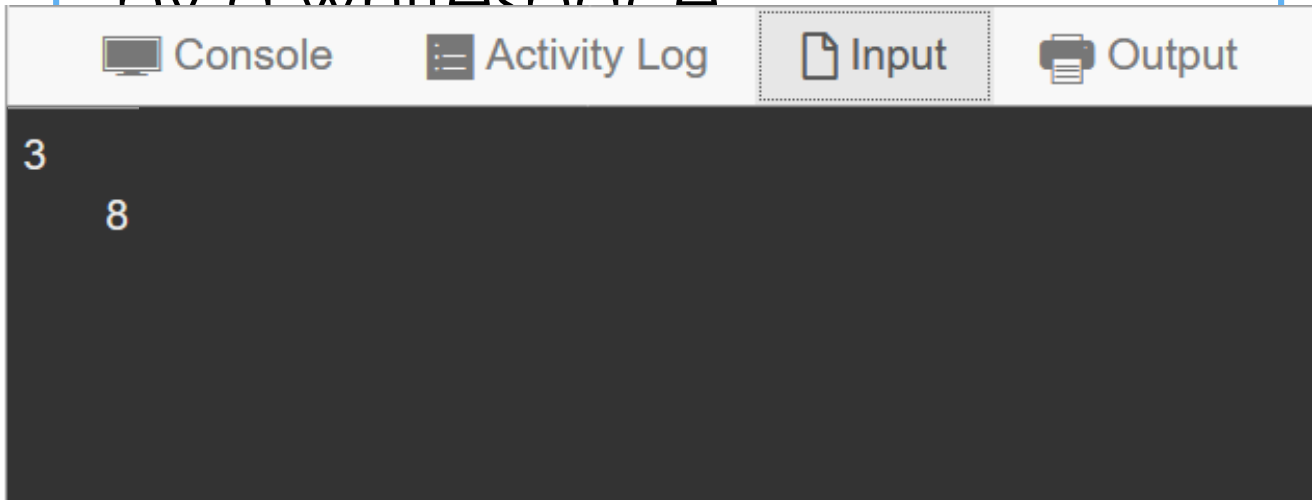
HOW WE USUALLY SPEAK TO A

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace



Shorthand for multiple inputs

HOW WE MUST SPEAK TO MR. COMPILER

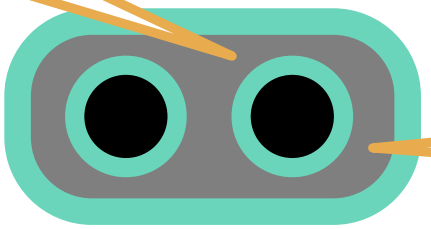
```
#include <stdio.h>
int main(){
int a, b;
                d", &a, &b);
                + b);
return 0;
}
```

3 8

a b

11

All look the same to me



11

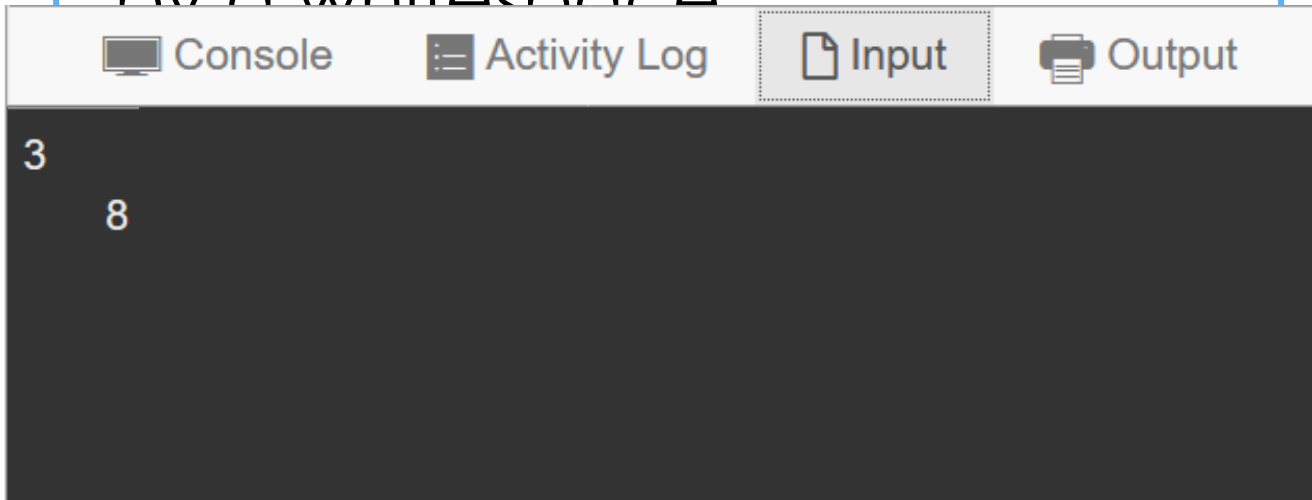
HOW WE USUA

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace



Shorthand for multiple inputs

Help!!!

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
int a, b;
d", &a, &b);
 + b);
return 0;
}
```

3 → a

8 → b

11

All look the same to me

11

Okay okay

HOW WE USUA

Do you speak English?

Hello

a, b are two variables.

Ask me for values of a and b. I will give both values separated by a whitespace

Console | Activity Log | Input | Output

```
3
8
```

What is going on with scanf?

What is going on with scanf?

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

What is going on with scanf?

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

Scanf follows this exact same rule while telling Mr. C how to read

What is going on with scanf?

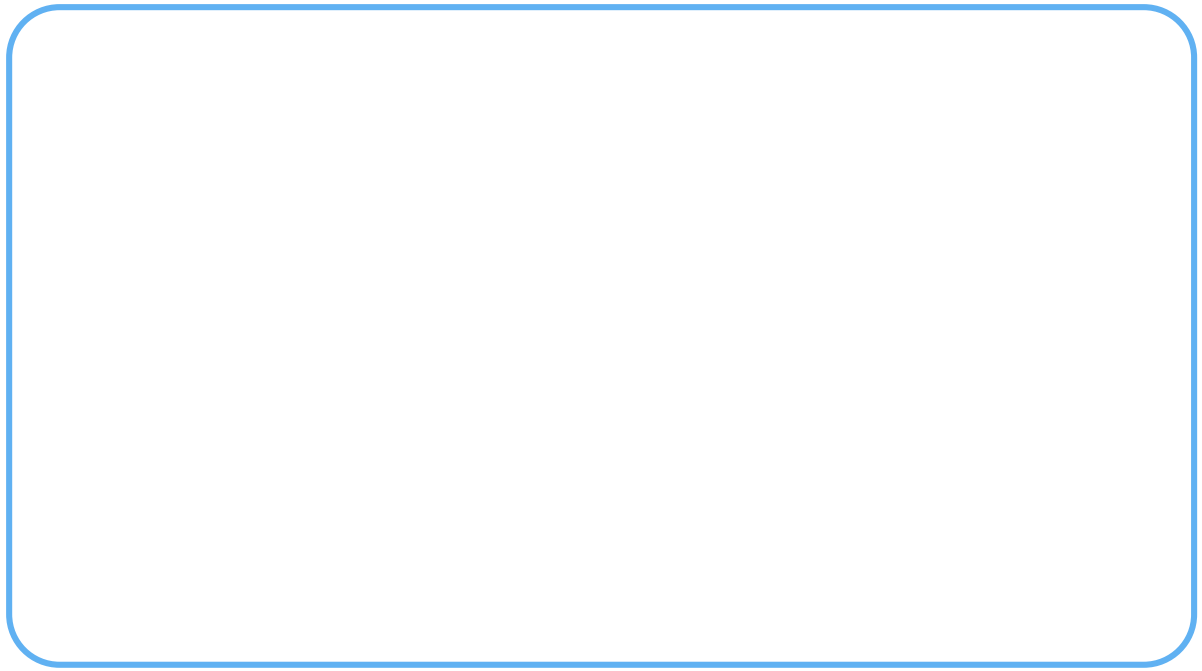
HOW WE USUALLY SPEAK TO A HUMAN

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

Scanf follows this exact same rule while telling Mr. C how to read

What is going on with scanf?

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!

Scanf follows this exact same rule while telling Mr. C how to read

What is going on with scanf?

HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

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

Scanf follows this exact same rule while telling Mr. C how to read

What is going on with scanf?

HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

Store value of the first integer in a and value of second integer in b.

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

Scanf follows this exact same rule while telling Mr. C how to read

What is going on with scanf?

HOW WE MUST SPEAK TO MR. COMPILER

HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

Store value of the first integer in a and value of second integer in b.

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

Scanf follows this exact same rule while telling Mr. C how to read

What is going on with scanf?

HOW WE MUST SPEAK TO MR. COMPILER

HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

Store value of the first integer in a and value of second integer in b.

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

Scanf follows this exact same rule while telling Mr. C how to read

What is going on with scanf?

HOW WE MUST SPEAK TO MR. COMPILER

```
scanf("%d%d", &a, &b);
```

HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

Store value of the first integer in a and value of second integer in b.

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

Scanf follows this exact same rule while telling Mr. C how to read

What is going on with scanf?

HOW WE MUST SPEAK TO MR. COMPILER

```
scanf("%d%d", &a, &b);
```

HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

Store value of the first integer in a and value of second integer in b.

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

Scanf follows this exact same rule while telling Mr. C how to read

What is going on with scanf?

HOW WE MUST SPEAK TO MR. COMPILER

```
scanf("%d%d", &a, &b);
```

HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

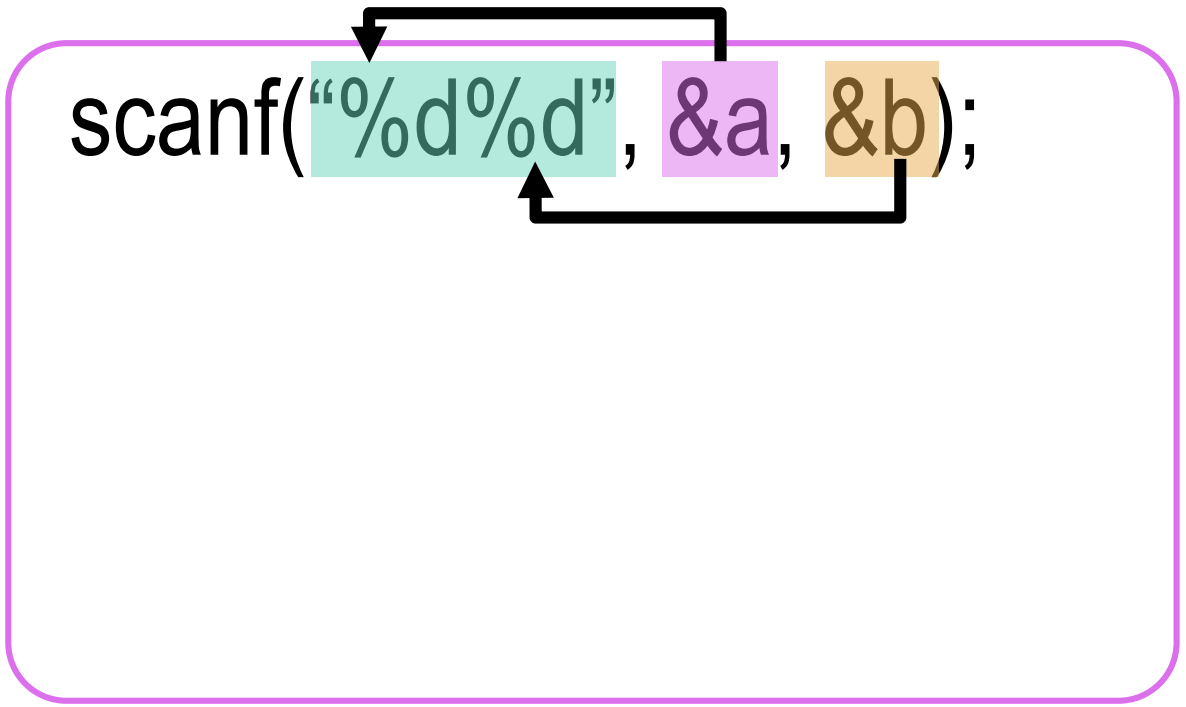
Store value of the first integer in a and value of second integer in b.

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

Scanf follows this exact same rule while telling Mr. C how to read

What is going on with scanf?

HOW WE MUST SPEAK TO MR. COMPILER



HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

Store value of the first integer in a and value of second integer in b.

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

Scanf follows this exact same rule while telling Mr. C how to read

What is going on with scanf?

HOW WE MUST SPEAK TO MR. COMPILER

```
scanf("%d%d", &a, &b);
```

Format string

HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

Store value of the first integer in a and value of second integer in b.

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

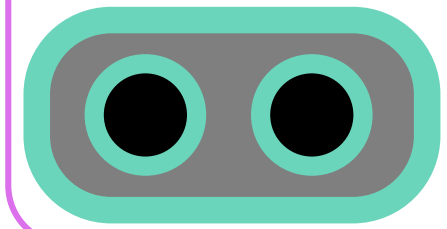
Scanf follows this exact same rule while telling Mr. C how to read

What is going on with scanf?

HOW WE MUST SPEAK TO MR. COMPILER

```
scanf("%d%d", &a, &b);
```

Format string



HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

Store value of the first integer in a and value of second integer in b.

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

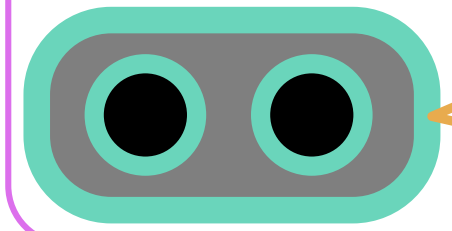
Scanf follows this exact same rule while telling Mr. C how to read

What is going on with scanf?

HOW WE MUST SPEAK TO MR. COMPILER

```
scanf("%d%d", &a, &b);
```

Format string



Format string tells me **how** you will write things, and then I am told **where** to store what I have read

HOW WE USUALLY SPEAK TO A HUMAN

Please read one integer. Ignore all whitespace (spaces,tabs,newlines) after that till I write another integer. Read that second integer too.

Store value of the first integer in a and value of second integer in b.

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

Scanf follows this exact same rule while telling Mr. C how to read

What is going on with scanf?

What is going on with scanf?

Be a bit careful since Mr C is a bit careless in this matter

What is going on with scanf?

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

What is going on with scanf?

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

scanf will never print anything

What is going on with scanf?

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

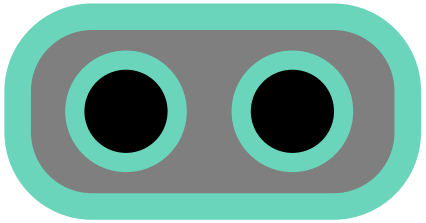
scanf will never print anything

```
scanf("Hello %d",&a);
```

What is going on with scanf?

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input



scanf will never print anything

```
scanf("Hello %d",&a);
```

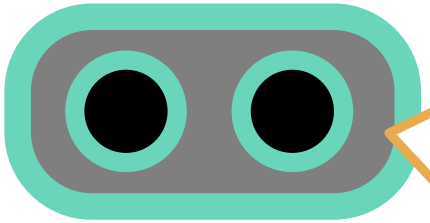
What is going on with scanf?

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

scanf will never print anything

```
scanf("Hello %d",&a);
```



Hmm … you are going to write the English word Hello followed by space followed by an integer. I will store the value of that integer in a

What is going on with scanf?

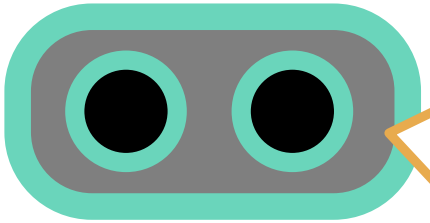
Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

scanf will never print anything

```
scanf("Hello %d",&a);
```

Use printf to print and scanf to read



Hmm … you are going to write the English word Hello followed by space followed by an integer. I will store the value of that integer in a

What is going on with scanf?

Be a bit careful since Mr C is a bit careless in this matter

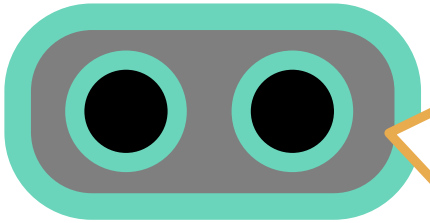
He treats all whitespace characters the same when integers are being input

scanf will never print anything

```
scanf("Hello %d",&a);
```

Use printf to print and scanf to read

Try out what happens with the following



Hmm … you are going to write the English word Hello followed by space followed by an integer. I will store the value of that integer in a

What is going on with scanf?

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

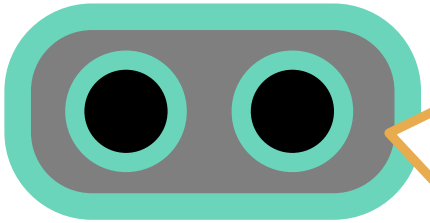
scanf will never print anything

```
scanf("Hello %d",&a);
```

Use printf to print and scanf to read

Try out what happens with the following

```
scanf("%d %d",&a,&b);
```

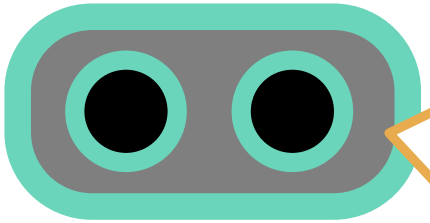


Hmm … you are going to write the English word Hello followed by space followed by an integer. I will store the value of that integer in a

What is going on with scanf?

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input



Hmm … you are going to write the English word Hello followed by space followed by an integer. I will store the value of that integer in a

scanf will never print anything

```
scanf("Hello %d",&a);
```

Use printf to print and scanf to read

Try out what happens with the following

```
scanf("%d %d",&a,&b);
scanf("%d,%d",&a,&b);
```

What is going on with scanf?

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

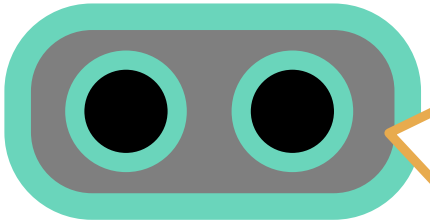
scanf will never print anything

```
scanf("Hello %d",&a);
```

Use printf to print and scanf to read

Try out what happens with the following

```
scanf("%d %d",&a,&b);
scanf("%d,%d",&a,&b);
scanf("%d\n%d",&a,&b);
```



Hmm … you are going to write the English word Hello followed by space followed by an integer. I will store the value of that integer in a

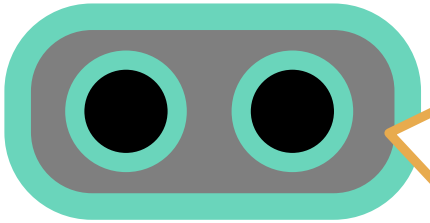
What is going on with scanf?

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

scanf will never print anything

```
scanf("Hello %d",&a);
```



Hmm … you are going to write the English word Hello followed by space followed by an integer. I will store the value of that integer in a

Use printf to print and scanf to read

Try out what happens with the following

```
scanf("%d %d",&a,&b);
scanf("%d,%d",&a,&b);
scanf("%d\n%d",&a,&b);   scanf("%d\t%d",&a,&b);
```

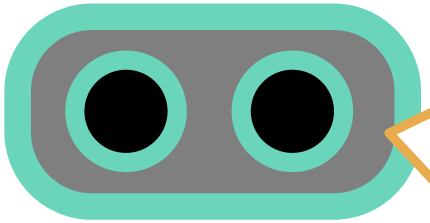
What is going on with scanf?

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

scanf will never print anything

```
scanf("Hello %d",&a);
```



Hmm … you are going to write the English word Hello followed by space followed by an integer. I will store the value of that integer in a

Use printf to print and scanf to read

Try out what happens with the following

```
scanf("%d %d",&a,&b);
scanf("%d,%d",&a,&b);
scanf("%d\n%d",&a,&b);
scanf("\"%d%d\"",&a,&b);
scanf("%d\t%d",&a,&b);
```

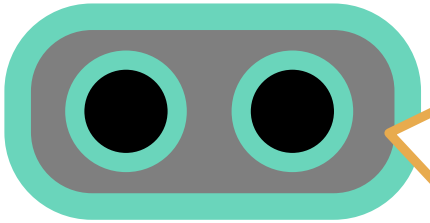
What is going on with scanf?

Be a bit careful since Mr C is a bit careless in this matter

He treats all whitespace characters the same when integers are being input

scanf will never print anything

```
scanf("Hello %d",&a);
```



Hmm … you are going to write the English word Hello followed by space followed by an integer. I will store the value of that integer in a

Use printf to print and scanf to read

Try out what happens with the following

```
scanf("%d %d",&a,&b);
scanf("%d,%d",&a,&b);
scanf("%d\n%d",&a,&b);
scanf("%dHello%d",&a,&b);
scanf("\"%d%d\"",&a,&b);
scanf("%d\t%d",&a,&b);
```

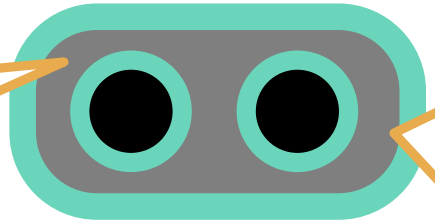
What is going on with scanf?

Be a bit careful since Mr C is a bit careless in this matter

He trea haracters the same when integer

scanf v ing

My advice to you is to take input one at a time in the beginning ☺ Try out acrobatics in free time



Hmm … you are going to write the English word Hello followed by space followed by an integer. I will store the value of that integer in a

```
scanf("Hello %d",&a);
```

Use printf to print and scanf to read

Try out what happens with the following

```
scanf("%d %d",&a,&b);
scanf("%d,%d",&a,&b);
scanf("%d\n%d",&a,&b);
scanf("%dHello%d",&a,&b);
scanf("\"%d%d\"",&a,&b);
scanf("%d\t%d",&a,&b);
```

What's in a Name?

What's in a Name?

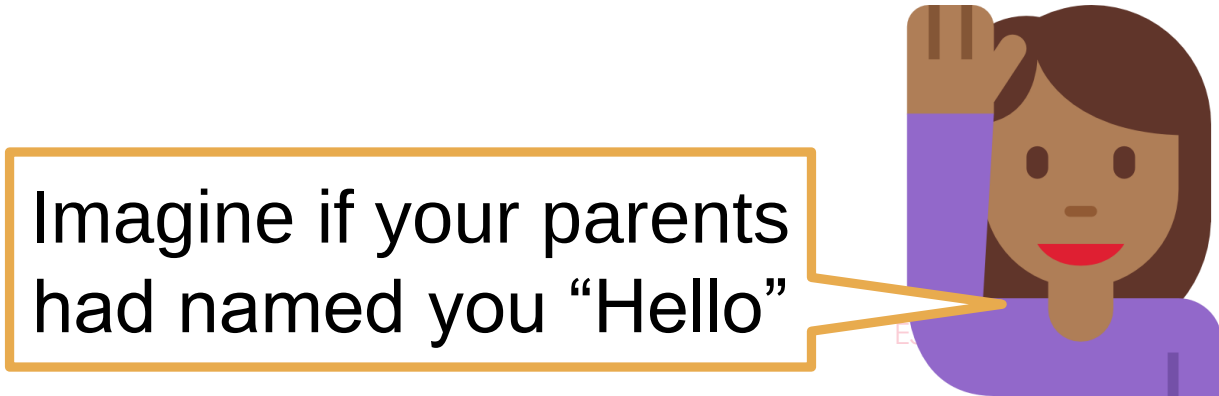
- Mr. C really does not care what we name our variables so long as they are legal names

What's in a Name?

- Mr. C really does not care what we name our variables so long as they are legal names

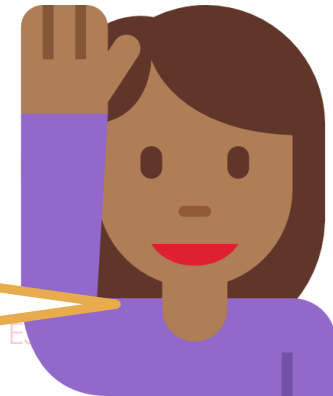
What's in a Name?

- Mr. C really does not care what we name our variables so long as they are legal names



What's in a Name?

- Mr. C really does not care what we name our variables so long as they are legal names

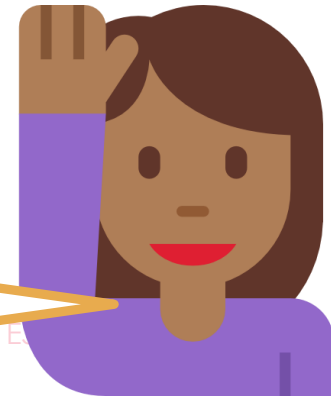


What's in a Name?

- Mr. C really does not care what we name our variables so long as they are legal names

My friends would greet me as Hello Hello?

Imagine if your parents had named you "Hello"



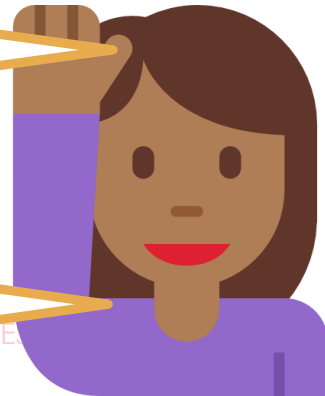
What's in a Name?

- Mr. C really does not care what we name our variables so long as they are legal names

My friends would greet me as Hello Hello?

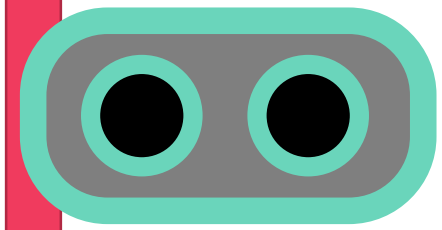
Wouldn't be nice, right? Mr C thinks so too.

Imagine if your parents had named you "Hello"



What's in a Name?

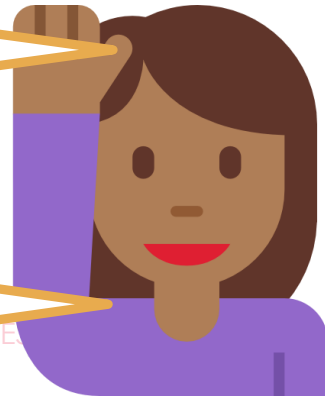
- Mr. C really does not care what we name our variables so long as they are legal names



My friends would greet me as Hello Hello?

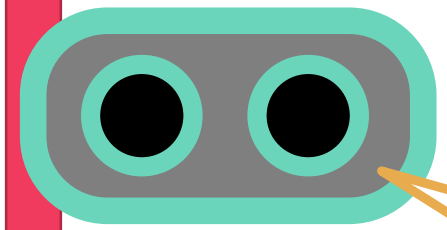
Wouldn't be nice, right? Mr C thinks so too.

Imagine if your parents had named you "Hello"



What's in a Name?

- Mr. C really does not care what we name our variables so long as they are legal names

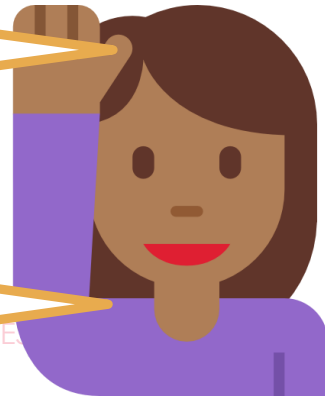


She's right. Some variable names are illegal

My friends would greet me as Hello Hello?

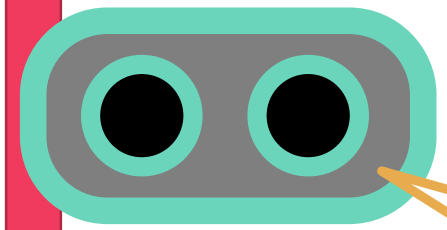
Wouldn't be nice, right? Mr C thinks so too.

Imagine if your parents had named you "Hello"



What's in a Name?

- Mr. C really does not care what we name our variables so long as they are legal names
- Certain names are called "reserved keywords"

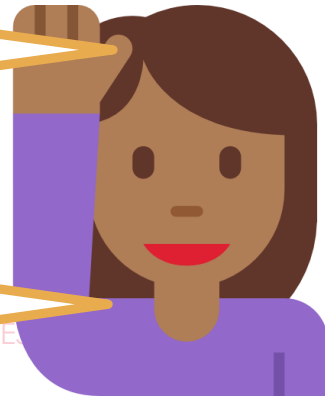


She's right. Some variable names are illegal

My friends would greet me as Hello Hello?

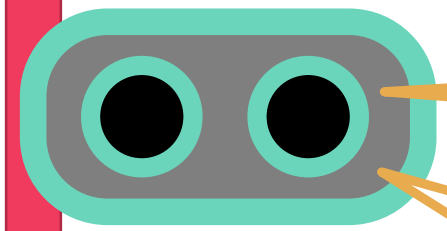
Wouldn't be nice, right? Mr C thinks so too.

Imagine if your parents had named you "Hello"



What's in a Name?

- Mr. C really does not care what we name our variables so long as they are legal names
- Certain names are called "reserved keywords"



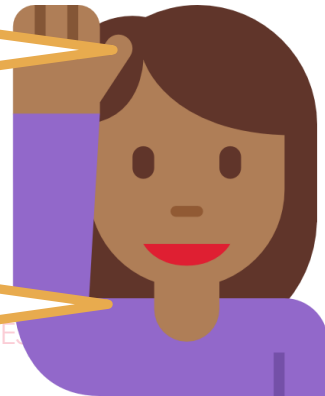
Cannot use reserved keywords as variable names

She's right. Some variable names are illegal

Wouldn't be nice, right? Mr C thinks so too.

My friends would greet me as Hello Hello?

Imagine if your parents had named you "Hello"



Reserved Keywords

Reserved Keywords

auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, return, short, signed, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile, while

Reserved Keywords

auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, return, short, signed, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile, while

Prutor shows keywords in a different color ☺

Reserved Keywords

auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, return, short, signed, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile, while

Prutor shows keywords in a different color ☺

Have already seen int and return in use

Reserved Keywords

auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, return, short, signed, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile, while

Prutor shows keywords in a different color ☺

Have already seen int and return in use

Will soon see use for long, float

Reserved Keywords

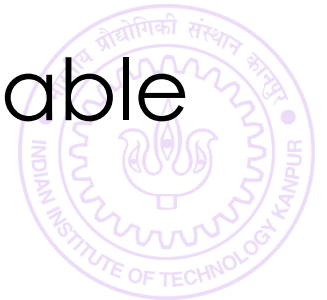
auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, return, short, signed, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile, while

Prutor shows keywords in a different color ☺

Have already seen int and return in use

Will soon see use for long, float

By the end of course, you will become very comfortable with most of these keywords ☺



How to give Names in C

How to give Names in C

Mr C calls names we give to our variables as *identifiers*

How to give Names in C

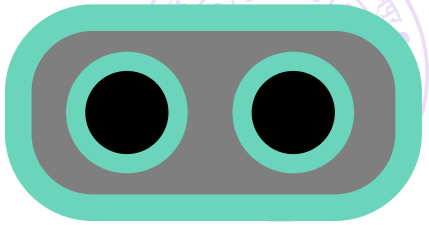
Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

How to give Names in C

Mr C calls names we give to our variables as *identifiers*

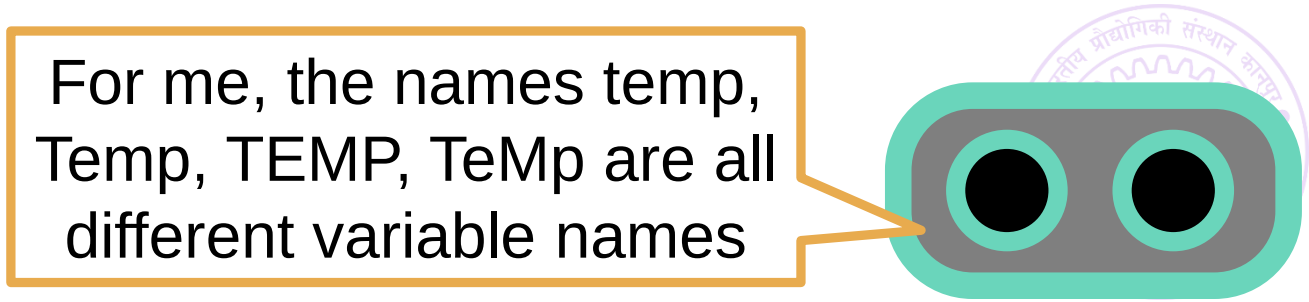
Names are case-sensitive



How to give Names in C

Mr C calls names we give to our variables as *identifiers*

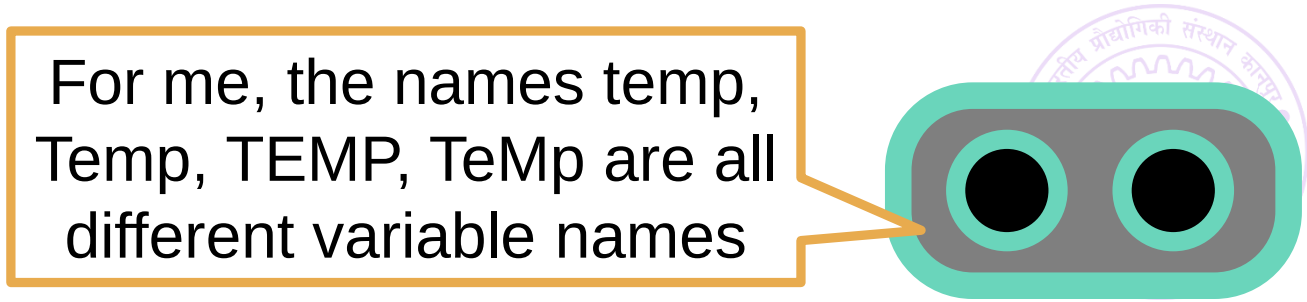
Names are case-sensitive



How to give Names in C

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive



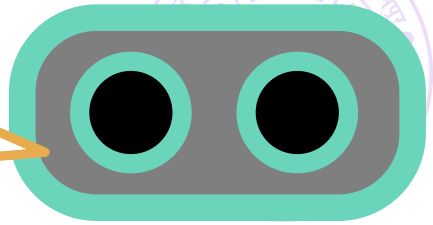
How to give Names in C

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

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

For me, the names temp, Temp, TEMP, TeMp are all different variable names



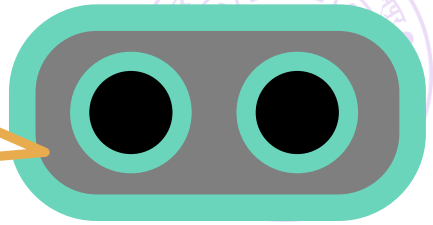
How to give Names in C

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

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

For me, the names temp, Temp, TEMP, TeMp are all different variable names



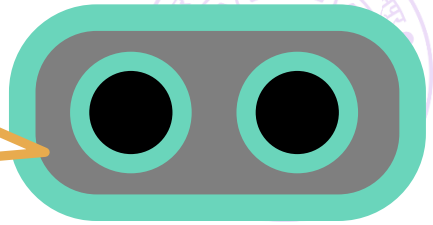
How to give Names in C

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

```
#include <stdio.h>
int main(){
int temp, TEMP, Temp, TeMp;
```

For me, the names temp, Temp, TEMP, TeMp are all different variable names



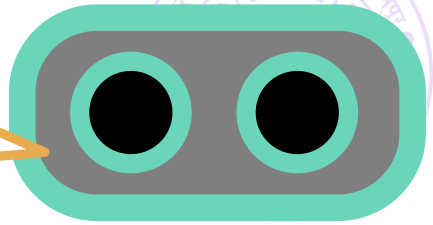
How to give Names in C

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

```
#include <stdio.h>
int main(){
int temp, TEMP, Temp, TeMp;
return 0;
```

For me, the names temp, Temp, TEMP, TeMp are all different variable names



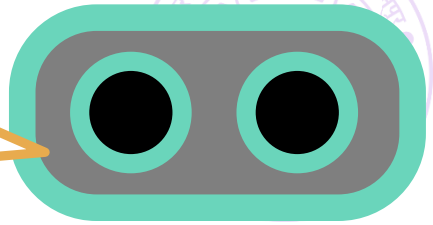
How to give Names in C

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

```
#include <stdio.h>
int main(){
int temp, TEMP, Temp, TeMp;
return 0;
}
```

For me, the names temp, Temp, TEMP, TeMp are all different variable names



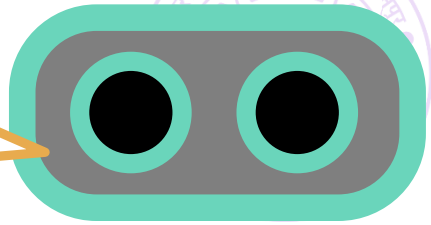
How to give Names in C

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

```
#include <stdio.h>
int main(){
int temp, TEMP, Temp, TeMp;
return 0;
}
```

For me, the names temp, Temp, TEMP, TeMp are all different variable names

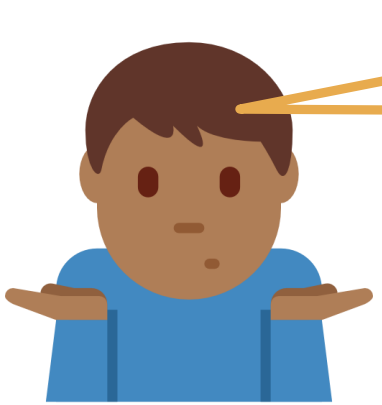


How to give Names in C

Mr C calls names we give to our variables as *identifiers*

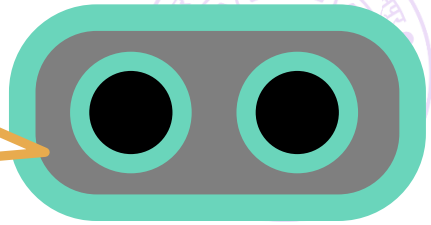
Names are case-sensitive

```
#include <stdio.h>
int main(){
int temp, TEMP, Temp, TeMp;
return 0;
}
```



So this program is fine?

For me, the names temp, Temp, TEMP, TeMp are all different variable names

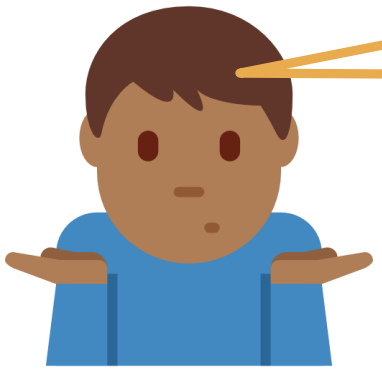


How to give Names in C

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

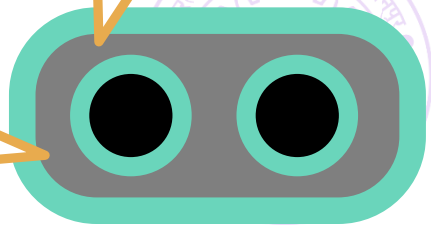
```
#include <stdio.h>
int main(){
int temp, TEMP, Temp, TeMp;
return 0;
}
```



So this program is fine?

For me, the names temp, Temp, TEMP, TeMp are all different variable names

Yes, but not advisable. May make mistakes, confuse others



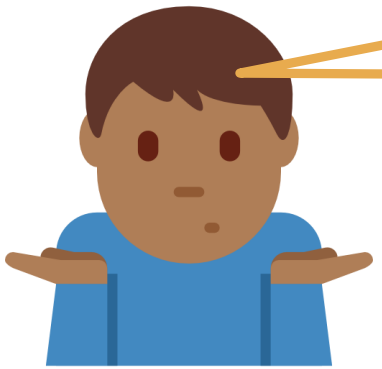
How to give Names in C

Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

Names can contain A-Z, a-z, 0-9, underscore _

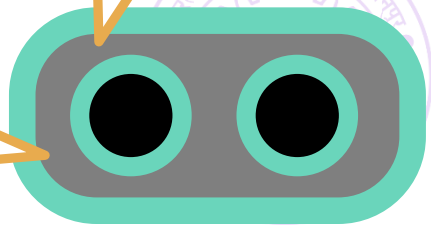
```
#include <stdio.h>
int main(){
int temp, TEMP, Temp, TeMp;
return 0;
}
```



So this program is fine?

For me, the names temp, Temp, TEMP, TeMp are all different variable names

Yes, but not advisable. May make mistakes, confuse others



How to give Names in C

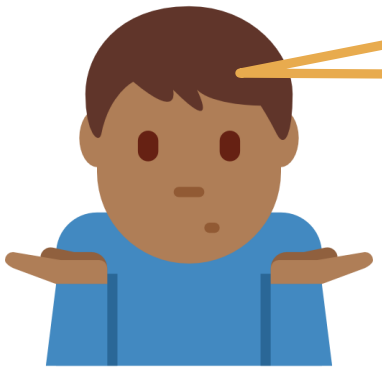
Mr C calls names we give to our variables as *identifiers*

Names are case-sensitive

Names can contain A-Z, a-z, 0-9, underscore _

Names cannot start with a digit

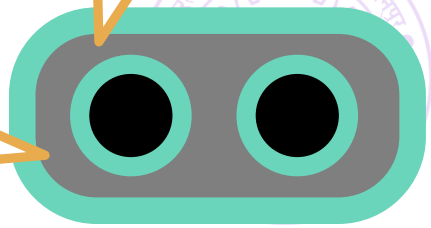
```
#include <stdio.h>
int main(){
int temp, TEMP, Temp, TeMp;
return 0;
}
```



So this program is fine?

For me, the names temp, Temp, TEMP, TeMp are all different variable names

Yes, but not advisable. May make mistakes, confuse others



Writing pretty code is an art

Writing pretty code is an art


```
1iitk,me@iitk,#iitk,iitk.1,
        iitk-1, iit k,
```

Writing pretty code is an art

1iitk,me@iitk,#iitk,iitk.1,
iitk-1, iit k,

Writing pretty code is an art

1iitk,me@iitk,#iitk,iitk.1,
iitk-1, iit k,

iitk1,iitk_1,me_iitk,
iitk_one, iitK

Writing pretty code is an art

1iitk,me@iitk,#iitk,iitk.1,
iitk-1, iit k,

iitk1,iitk_1,me_iitk,
iitk_one, iitK

Writing pretty code is an art

1iitk,me@iitk,#iitk,iitk.1,
iitk-1, iit k,

iitk1,iitk_1,me_iitk,
iitk_one, iitK

Make names meaningful i.e. hzoxys valid name but ☹

Writing pretty code is an art

1iitk,me@iitk,#iitk,iitk.1,
iitk-1, iit k,

iitk1,iitk_1,me_iitk,
iitk_one, iitK

Make names meaningful i.e. hzoxys valid name but ☹

Programmers are artists and differ on what is more pretty

Writing pretty code is an art

1iitk,me@iitk,#iitk,iitk.1,
iitk-1, iit k,

iitk1,iitk_1,me_iitk,
iitk_one, iitK

Make names meaningful i.e. hzoxys valid name but ☹

Programmers are artists and differ on what is more pretty

numFirst, numSecond

Writing pretty code is an art

1iitk,me@iitk,#iitk,iitk.1,
iitk-1, iit k,

iitk1,iitk_1,me_iitk,
iitk_one, iitK

Make names meaningful i.e. hzoxys valid name but ☹

Programmers are artists and differ on what is more pretty

numFirst, numSecond num_first, num_second

Writing pretty code is an art

1iitk,me@iitk,#iitk,iitk.1, iitk-1, iit k,

iitk1,iitk_1,me_iitk, iitk_one, iitK

Make names meaningful i.e. hzoxys valid name but ☹

Programmers are artists and differ on what is more pretty

numFirst, numSecond num_first, num_second

Make names not too long e.g. tmp vs temporary

Writing pretty code is an art

1iitk,me@iitk,#iitk,iitk.1, iitk-1, iit k,

iitk1,iitk_1,me_iitk, iitk_one, iitK

Make names meaningful i.e. hzoxys valid name but ☹

Programmers are artists and differ on what is more pretty

numFirst, numSecond num_first, num_second

Make names not too long e.g. tmp vs temporary

You will gradually get used to many artistic styles

Writing pretty code is an art

1iitk,me@iitk,#iitk,iitk.1, iitk-1, iit k,

iitk1,iitk_1,me_iitk, iitk_one, iitK

Make names meaningful i.e. hzoxys valid name but ☹

Programmers are artists and differ on what is more pretty

numFirst, numSecond num_first, num_second

Make names not too long e.g. tmp vs temporary

You will gradually get used to many artistic styles

tmp, iter, var, fun, ref

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

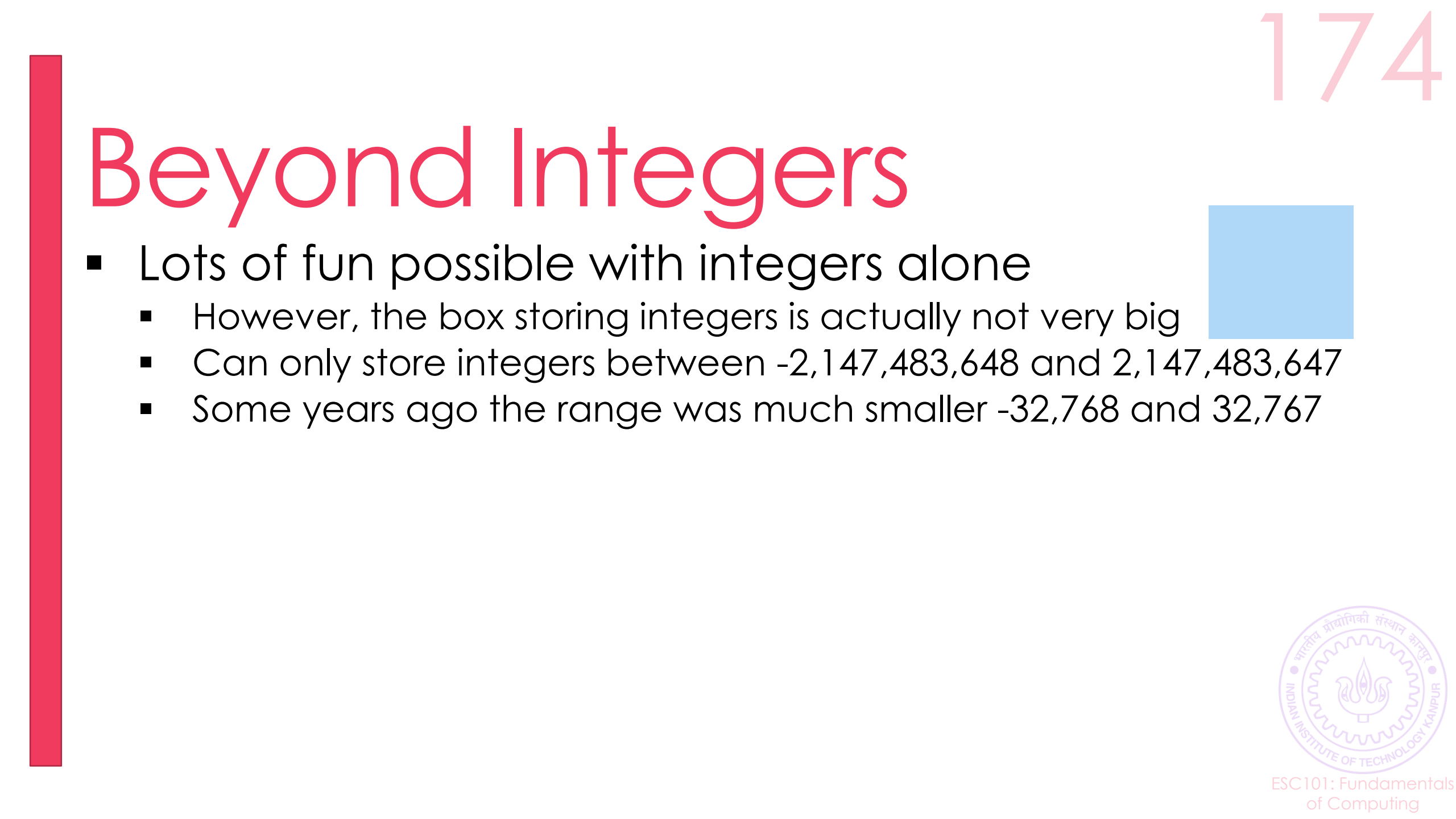
- Lots of fun possible with integers alone
  - However, the box storing integers is actually not very big

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
  - Some years ago the range was much smaller -32,768 and 32,767



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
  - Some years ago the range was much smaller -32,768 and 32,767
  - Computers have become really powerful in last 5-10 years

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
  - Some years ago the range was much smaller -32,768 and 32,767
  - Computers have become really powerful in last 5-10 years
- 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
  - Some years ago the range was much smaller -32,768 and 32,767
  - Computers have become really powerful in last 5-10 years
- 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
  - Some years ago the range was much smaller -32,768 and 32,767
  - Computers have become really powerful in last 5-10 years
- 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?

Long integers

Long integers

Really long – can store integers between

Long integers

Really long – can store integers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

Long integers

Really long – can store integers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

Long integers

Really long – can store integers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

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

Long integers

Really long – can store integers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

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

Long integers

Really long – can store integers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
long a; // long int a; works too!
```

Long integers

Really long – can store integers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
long a; // long int a; works too!
scanf("%ld", &a);
```

Long integers

Really long – can store integers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
long a; // long int a; works too!
scanf("%ld", &a);
printf("My first long int %ld", a);
```

Long integers

Really long – can store integers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
long a; // long int a; works too!
scanf("%ld", &a);
printf("My first long int %ld", a);
return 0;
```

Long integers

Really long – can store integers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
long a; // long int a; works too!
scanf("%ld", &a);
printf("My first long int %ld", a);
return 0;
}
```

Long integers

Really long – can store integers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
long a; // long int a; works too!
scanf("%ld", &a);
printf("My first long int %ld", a);
return 0;
}
```

Comment

Long integers

Really long – can store integers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
long a; // long int a; works too!
scanf("%ld", &a);
printf("My first long int %ld", a);
return 0;
}
```

Comment

%ld

Long integers

Really long – can store integers between

-9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
long a; // long int a; works too!
scanf("%ld", &a);
printf("My first long int %ld", a);
return 0;
}
```

Comment

%ld

Integer arithmetic applies to long int as well +, -, /, *, %, ()

Long integers

Really long – can store integers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
long a; // long int a; works too!
scanf("%ld", &a);
printf("My first long int %ld", a);
return 0;
}
```

Comment

%ld

Integer arithmetic applies to long int as well +, -, /, *, %, ()

Try them out on Prutor

Long integers

Really long – can store integers between

-9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
long a; // long int a; works too!
scanf("%ld", &a);
printf("My first long int %ld", a);
return 0;
}
```

Comment

%ld

Integer arithmetic applies to long int as well +, -, /, *, %, ()

Try them out on Prutor

How does long int interact with int – int + long, int * long?

Long integers

Really long – can store integers between

-9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

```
#include <stdio.h>
int main(){
long a; // long int a; works too!
scanf("%ld", &a);
printf("My first long int %ld", a);
return 0;
}
```

Comment

%ld

Integer arithmetic applies to long int as well +, -, /, *, %, ()

Try them out on Prutor

How does long int interact with int – int + long, int * long?

Next class ☺

How about Real numbers?

How about Real numbers?

int allows us to store, do math formulae with integers

How about Real numbers?

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

How about Real numbers?

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

How about Real numbers?

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

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

How about Real numbers?

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

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

How about Real numbers?

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

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

How about Real numbers?

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
float a;
scanf("%f", &a);
```

How about Real numbers?

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
float a;
scanf("%f", &a);
printf("My first real %f", a);
```

How about Real numbers?

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
float a;
scanf("%f", &a);
printf("My first real %f", a);
return 0;
```

How about Real numbers?

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
float a;
scanf("%f", &a);
printf("My first real %f", a);
return 0;
}
```

How about Real numbers?

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
float a;
scanf("%f", &a);
printf("My first real %f", a);
return 0;
}
```

How about Real numbers?

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
float a;
scanf("%f", &a);
printf("My first real %f", a);
return 0;
}
```

%f

Very large range 3.4e+38

How about Real numbers?

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
float a;
scanf("%f", &a);
printf("My first real %f", a);
return 0;
}
```

%f

Very large range 3.4e+38

Arithmetic operations apply to float as well +, -, /, *, ()

How about Real numbers?

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
float a;
scanf("%f", &a);
printf("My first real %f", a);
return 0;
}
```

%f

Very large range 3.4e+38

Arithmetic operations apply to float as well +, -, /, *, ()

How about Real numbers?

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

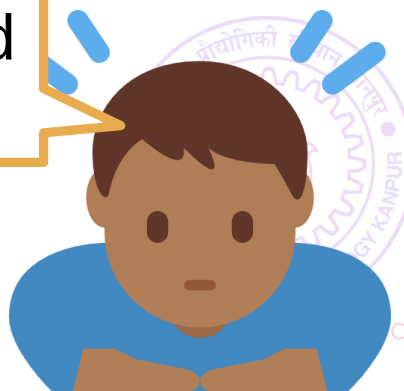
```
#include <stdio.h>
int main(){
float a;
scanf("%f", &a);
printf("My first real %f", a);
return 0;
}
```

%f

Very large range 3.4e+38

Arithmetic operations apply to float as well +, -, /, *, ()

What happened to %?



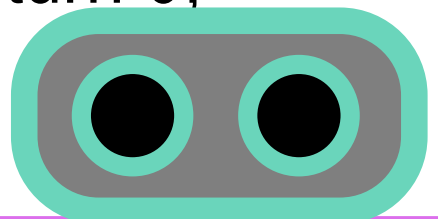
How about Real numbers?

int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

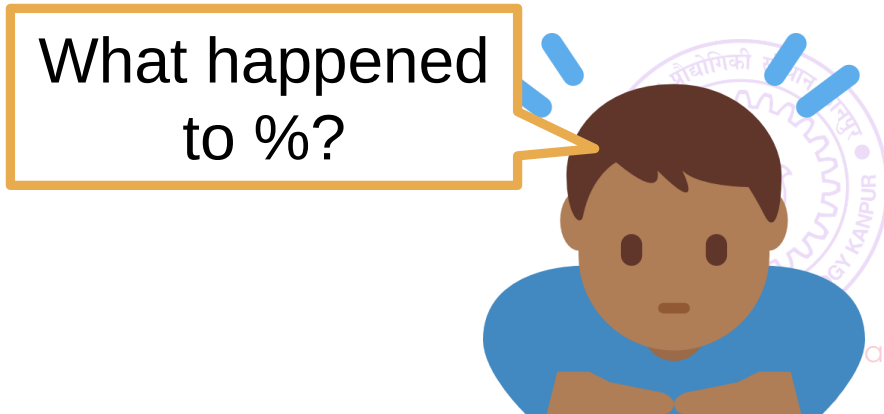
```
#include <stdio.h>
int main(){
float a;
scanf("%f", &a);
printf("My first real %f", a);
return 0;
}
```

%f



Very large range 3.4e+38

Arithmetic operations apply to float as well +, -, /, *, ()



How about Real numbers?

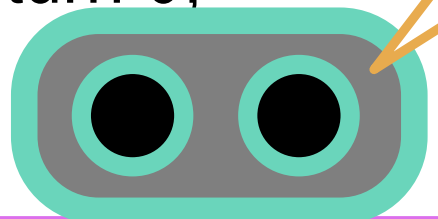
int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
float a;
scanf("%f", &a);
printf("My first rea
return 0;
}
```

%f

☺ Did you ever do remainders with real numbers in school?



Very large range 3.4e+38

Arithmetic operations apply to float as well +, -, /, *, ()

What happened to %?

How about Real numbers?

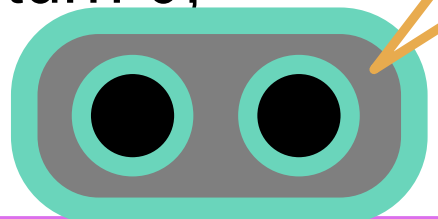
int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
float a;
scanf("%f", &a);
printf("My first rea
return 0;
}
```

%f

☺ Did you ever do remainders with real numbers in school?



Very large range 3.4e+38

Arithmetic operations apply to float as well +, -, /, *, ()

How about Real numbers?

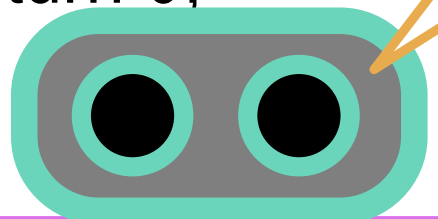
int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
float a;
scanf("%f", &a);
printf("My first rea
return 0;
}
```

%f

☺ Did you ever do remainders with real numbers in school?



Very large range 3.4e+38

Arithmetic operations apply to float as well +, -, /, *, ()

I remember. Remainders make sense for integers, not for real numbers

How about Real numbers?

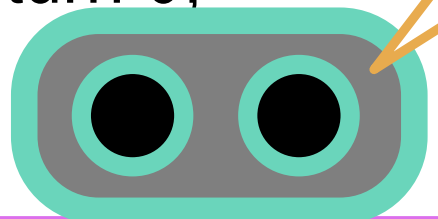
int allows us to store, do math formulae with integers

float allows us to store, do math formulae with reals

```
#include <stdio.h>
int main(){
float a;
scanf("%f", &a);
printf("My first rea
return 0;
}
```

%f

☺ Did you ever do remainders with real numbers in school?



Very large range 3.4e+38

Arithmetic operations apply to float as well +, -, /, *, ()

Try them out on Prutor

I remember. Remainders make sense for integers, not for real numbers