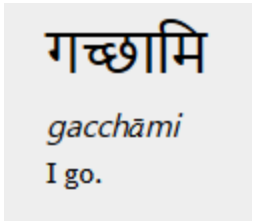
Basics of C Syntax, Printing Outputs (printf)

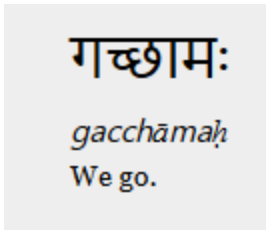
ESC101: Fundamentals of Computing

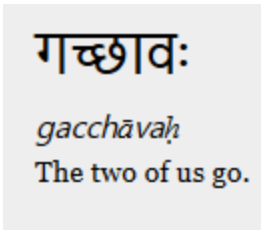
Nisheeth

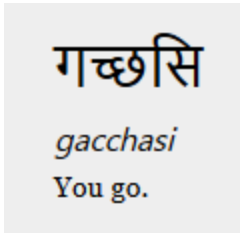
Syntax

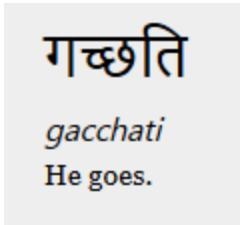
- The rules that decide how to combine symbols to form sentences
  - Subject-verb-object (SVO) languages, e.g. English, write sentences like, “the subject performed some verb upon the object”
  - Subject-object-verb (SOV) languages, e.g. Hindi, write sentences like, “subject ne object par kuchh verb kiya”











Syntax vs. standard

- If violating a rule makes the sentence nonsensical, you have violated a syntactic rule
- If violating a rule makes the sentence look unconventional, but still understandable, you have violated a standard

जानिए क्यों है आज भारत बंद,
ट्रेड यूनियनों की ये है डिमांड
लिस्ट

ईरान ने अमेरिकी
एयरबेस पर किया हमला,
दागे बैलिस्टिक मिसाइल

CAA-NRC प्रदर्शन पर बोलीं
दीपिका- हमारे देश की नींव
ऐसी नहीं रखी गई थी

Syntax vs. standard

- main() vs int main()
  - Some compilers will accept just using main, some will not
  - Some compilers will accept main functions that don't have a return statement
- We will follow the C11 standard in this course
  - Has been superseded by the C18 standard, but differences are minor and don't show up in the material covered in this course
- We may sometimes ask you questions for which the answer would be 'depends on the compiler version'

C Syntax: The "Alphabet" of C

- C programs can be written using the following alphabet
  - A B .... Z
  - a b .... z
  - 0 1 9
  - Space . , : ; ' $ "
  - # % & ! _ { } [ ] ( ) |
  - + - * / =

C Syntax: Variables and Constants

- Most C programs consist of variables or constants with some name

firstName, age, height, streetAddress, valueOfPi

More on naming conventions/rules later

- The value of each variables or constant is stored at some location in the computer's main memory (RAM)

- A variable's value can be changed during execution of the program

- A constant's value cannot be changed during execution of the program

C Syntax: C Keywords (root words)

- C language has a set of 32 keywords

Seen so far

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

- These keywords have special meaning

Prutor shows keywords in a different color

- Can't use keywords for other purposes (e.g., can't use them to declare variable names, constant values, or function names)

C Syntax: Keywords Usage

```
# include <stdio.h>
int main(){
   int main = 3;
   printf("%d", main);
   return 0;
}
```

- This WILL work
- Reason: main is not a keyword
- But not recommended

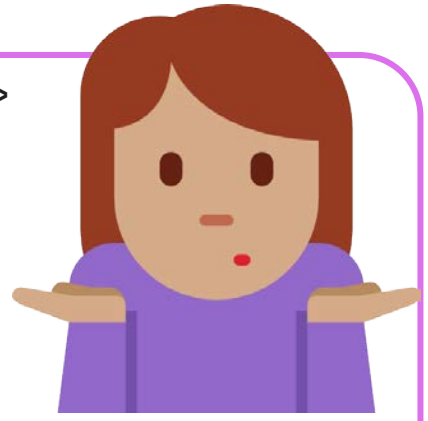
```
# include <stdio.h>
int main(){
   int return = 3;
   printf("%d", return);
   return 0;
}
```

- This will NOT work
- Reason: return is a keyword

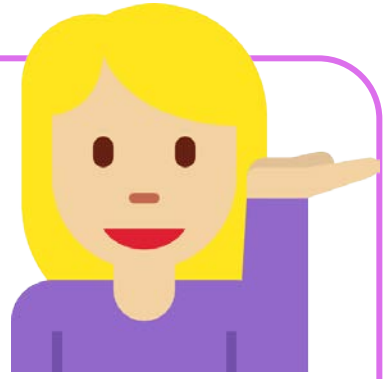
Advice: Try to Write Code that looks good

- Very important in industry - large groups collaborate
- Important even for solo projects - maintenance
- Will learn several good coding habits over time (such as Commenting, Indentation, Code-structuring)

```
#include<stdio.h>
int main(){
int a = 1;
int b = 2;
int c;
c = a + b;
printf("Result is %d",c);
return 0;
}
```



```
#include<stdio.h>
int main(){
    int a = 1;
    int b = 2;
    int c;
    c = a + b;
    printf("Result is %d",c);
    return 0;
}
```



This is good indentation

```
#include<stdio.h>
int main(){
 int a = 1;
          int b = 2;
    int c;
                                  ,c);
    }
```

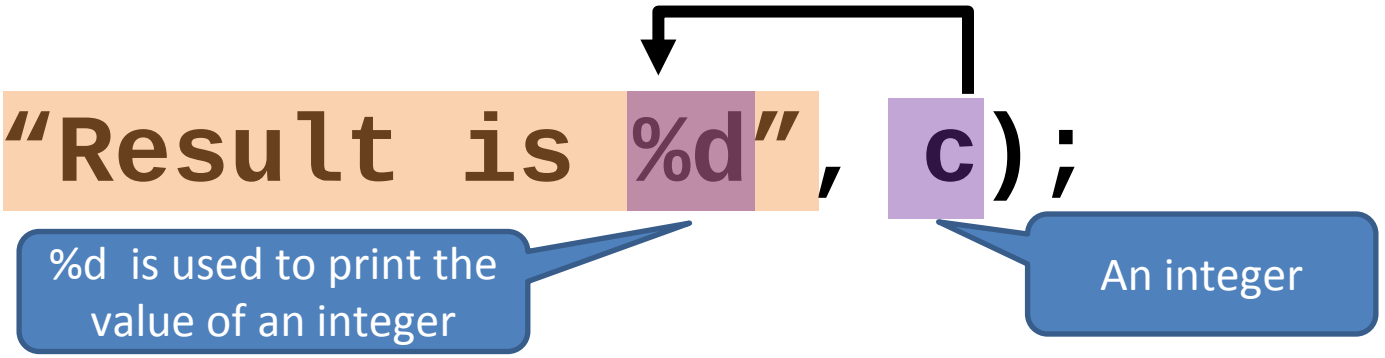
Prutor does it automatically when you write your code (try writing code in Prutor)

The printf Function

- A function used for printing the outputs of the C program
- Prints the outputs in a format specified by us
- We have already seen some simple examples of usage of printf

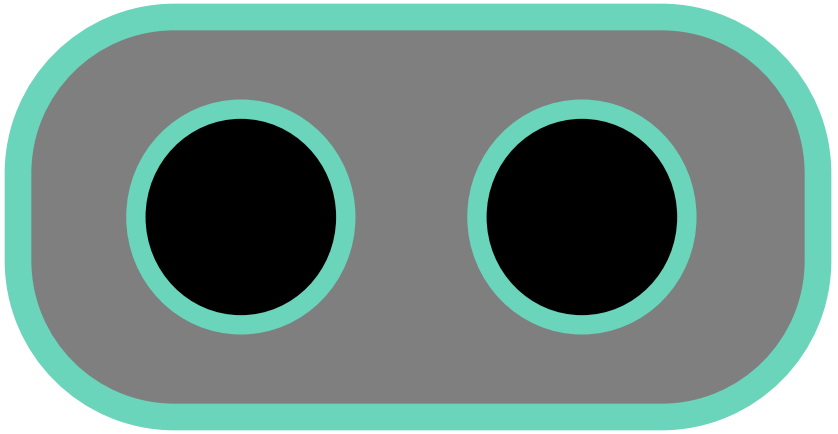
```
printf("Welcome to ESC101");
```

```
printf("Result is %d", c);
```



Introducing Mr. C (or Mr. Compiler)

- Will sometimes use this fictional character to refer to the C compiler (or our Prutor system)



- Sometimes it will mean the screen of the computer that shows us the program's output

True Power of printf

We have seen how to make Mr. C

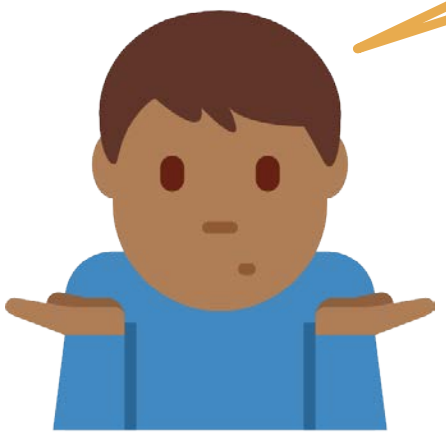
Say things like "Wel

Tell us the value of

Don't be afraid to experiment

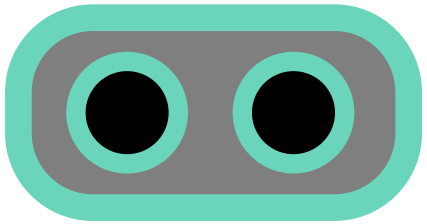
No, you can make him speak again and again

Can he speak only once?



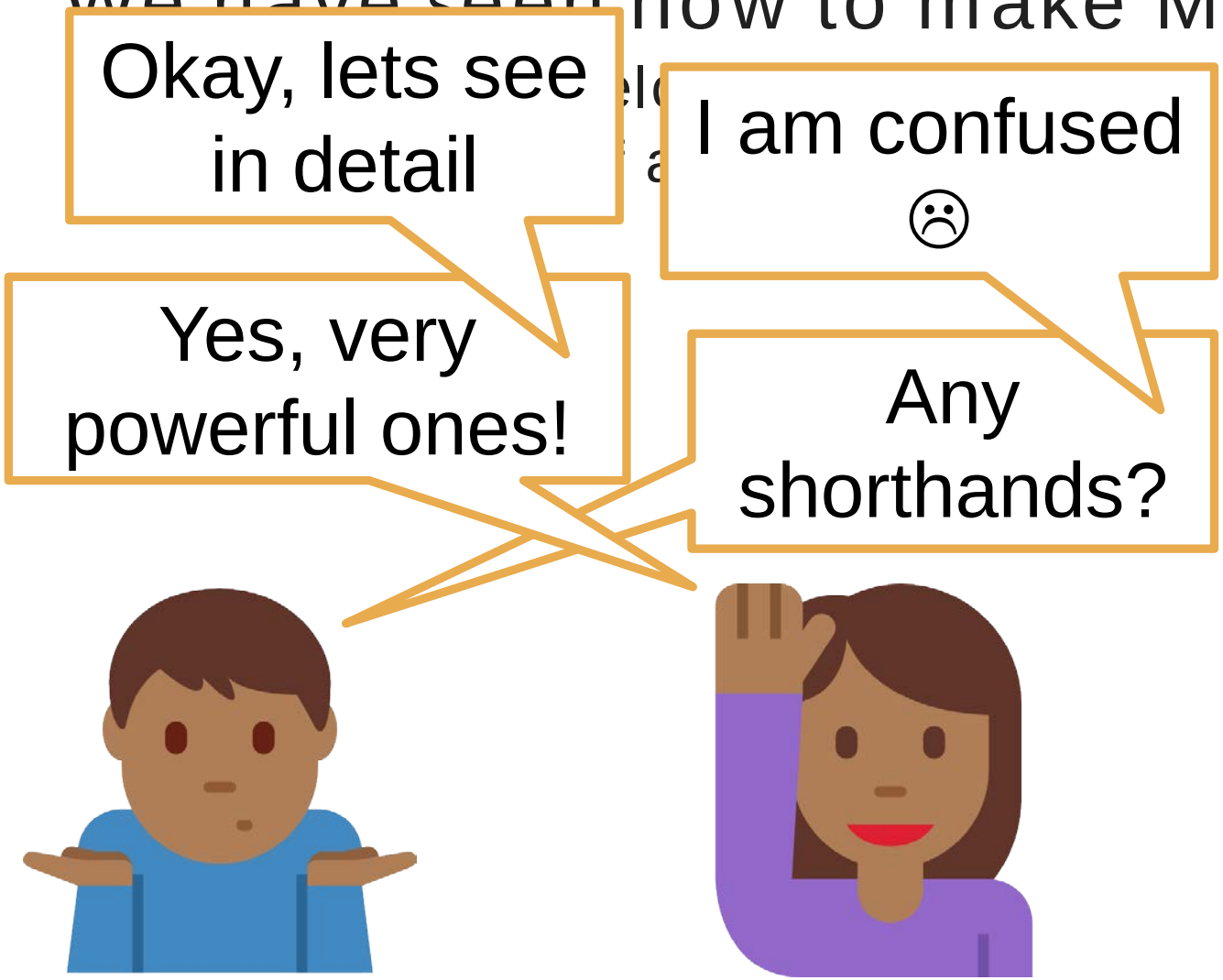
```
#include<stdio.h>
int main(){
int a = 5, b = 4;
printf("Hello ");
printf("%d",a);
printf("%d",b);
return 0;
}
```

Hello 54

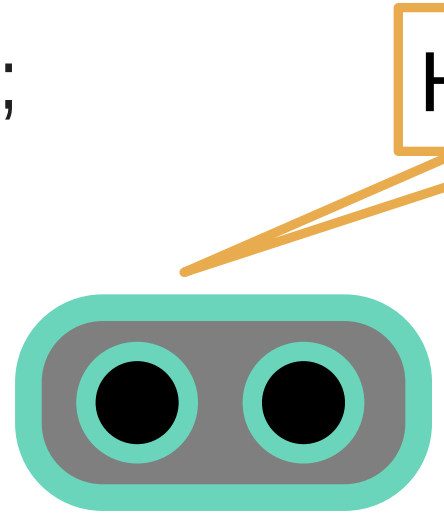


True Power of printf

We have seen how to make Mr. C




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

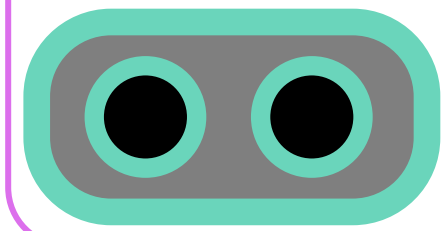


How printf Works?

HOW WE MUST SPEAK TO MR. COMPILER

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

Format string



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

HOW WE USUALLY SPEAK TO A HUMAN

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

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

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

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

True Power of printf

```
int main(){
    int a = 3;
    int b = 2;
    printf("Sum of a and b = %d", a+b);
   return 0;
}
```

Can also use printf to directly print the value of an expression

Summary: The syntax of printf

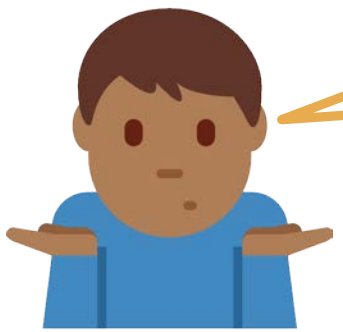
Note: In some cases, there will be no such list. Example: printf("Hello");

printf(format string, list of things to print);

Example (already seen)

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

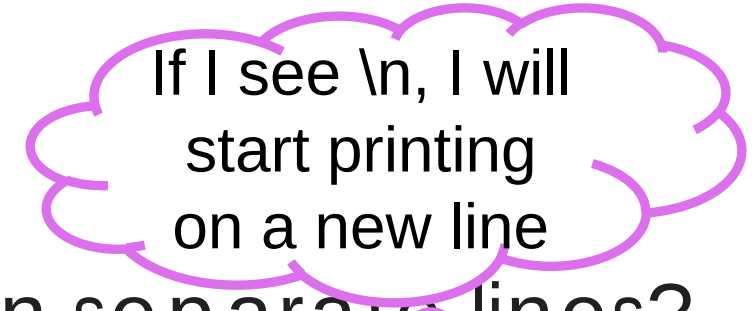
Important: Format string must have format specifiers for all things we wish to print in the exact same order as those things appear in the list of things



Can Mr. C print integers only?

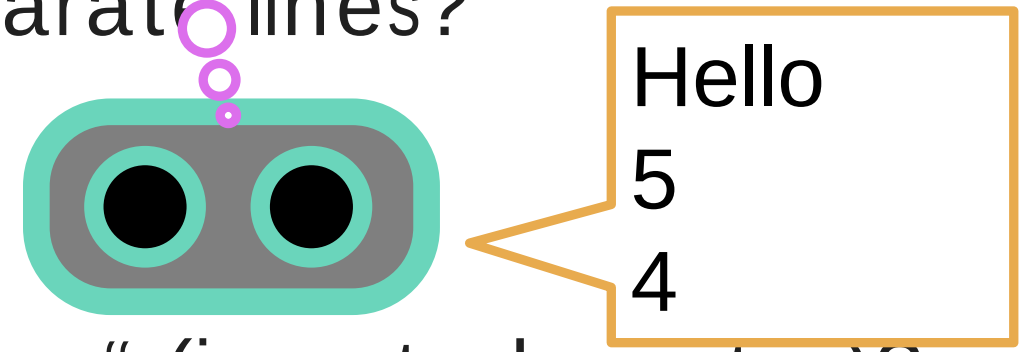
Of course NOT. Wait for next week's lectures

Some Fun with printf



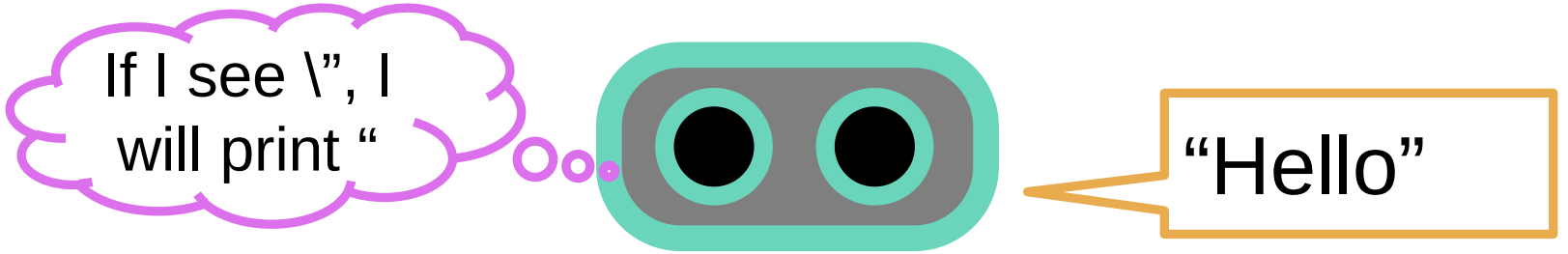
Can I print different things on separate lines?

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



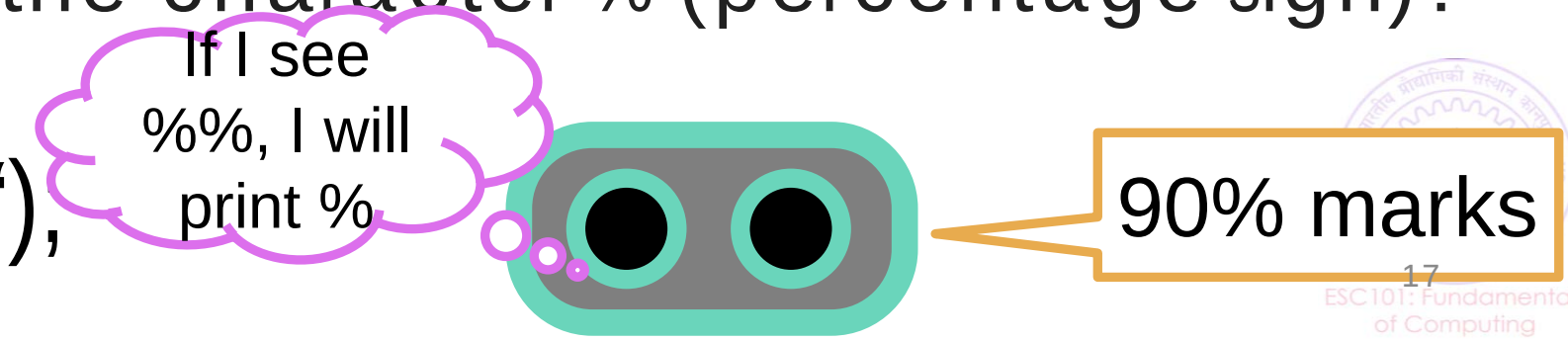
What if I wish to print the character " (inverted quotes)?

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



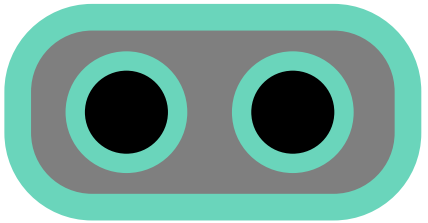
What if I wish to print the character % (percentage sign)?

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



Some Fun with printf

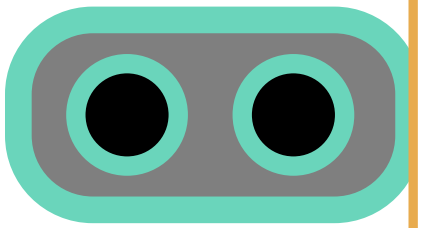
To print on new line, use \n



To print the character \ (backslash) use \\

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

To print a *tab* character (a lon



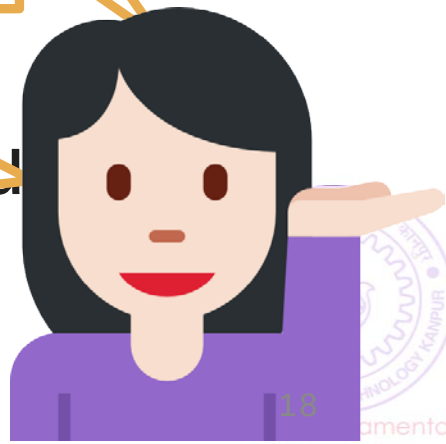
```
printf("Very\tNice");
```

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

Allows us to print very nicely formatted output ☺

More examples in labs – till

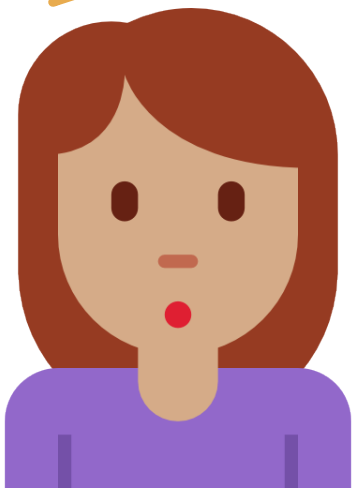
Experiment with them to get comfortable

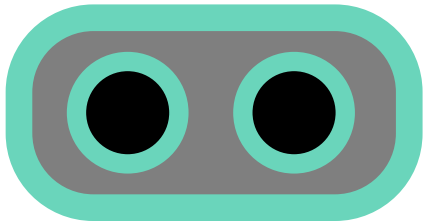


Fun with Integers

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

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





Not everything needs to be stored in a variable

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

printf("%d %d",a,10);
is fine too ☺

Come to the lab and give it a try!

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

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

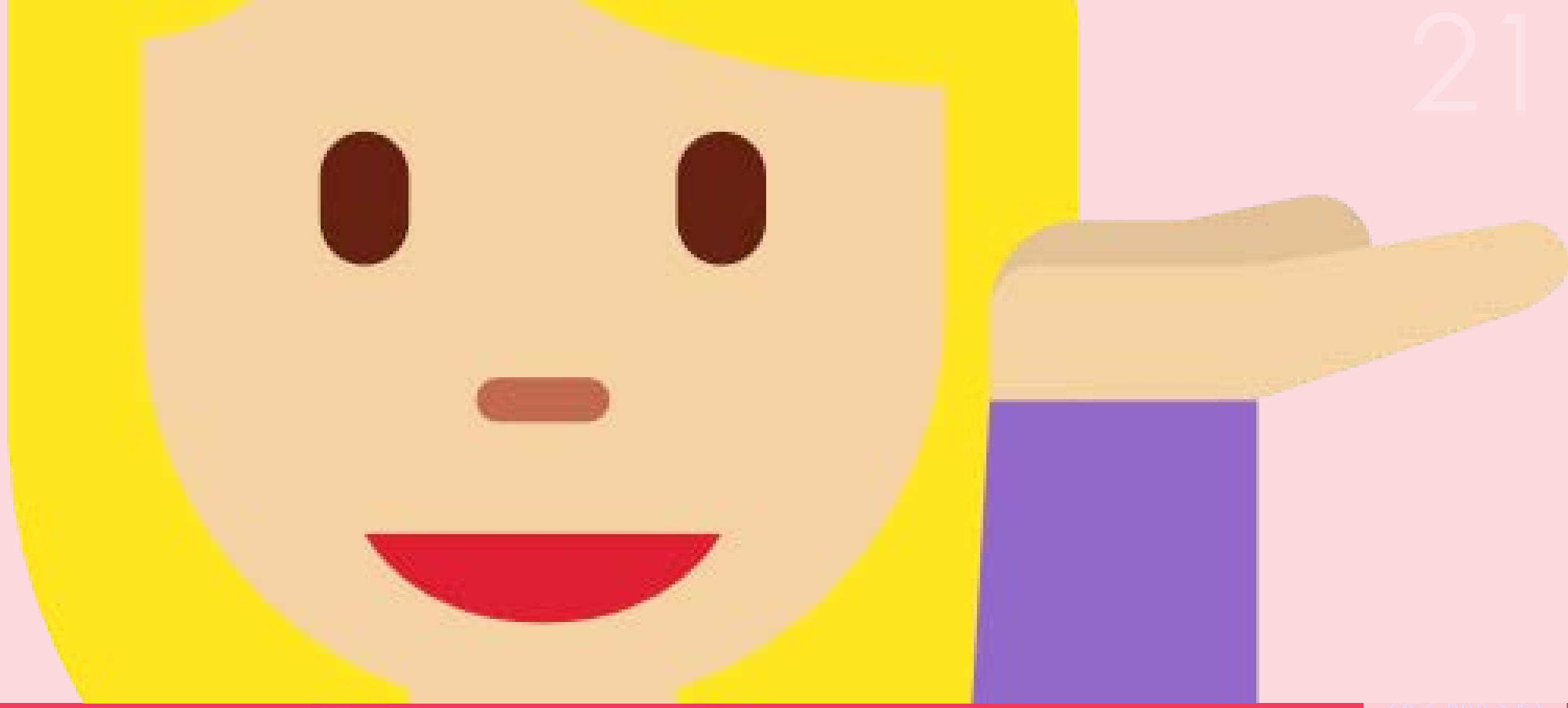
Also be careful about division and remainder

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

Experiment on your own – will revisit these very soon

Life beyond Integers

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



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