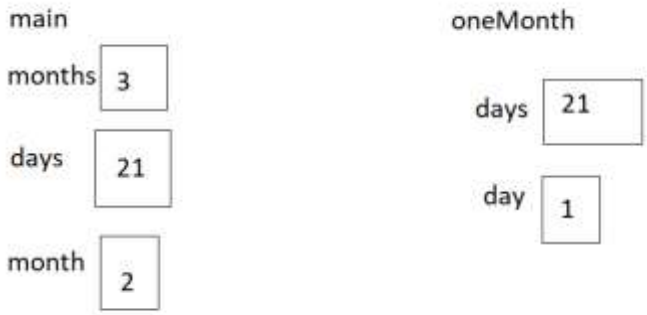
Calendar Example

First – just did a main function that prints one month.

Next – Changed main to be called oneMonth, added a new main which looped through the months and called oneMonth each time. Every time oneMonth is called, it did the same thing.

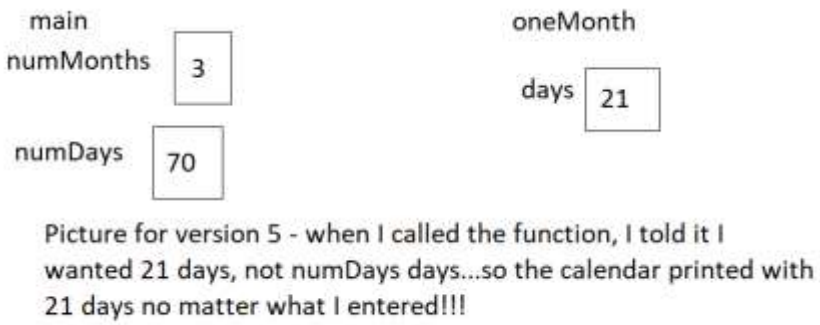
Third – It was strange that oneMonth always asked for the number of days. We should do that in main. But then, we need to "transfer" that information to the function. We do that via a parameter, adding the name of the variable in the parentheses when defining the function.



Fourth – To illustrate that each main and oneMonth has their own variables, we called the variables different names (days and numDays) and showed what happens when we call the function with the wrong name. In the video we showed how the VALUE of the actual parameter (in the function call), gets copied into the box for the formal parameter (in the function).

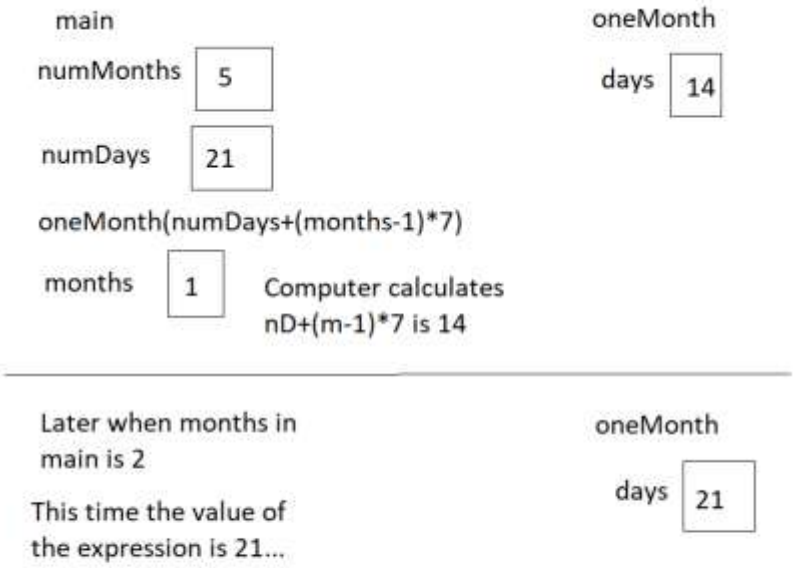
Picture of calendarfunc4 - the value of the actual parameter, numDays, gets copied into the box for the formal parameter, days.

Fifth – Showed what happens when we ignore the user input and always pass in 21 for # of days.



Picture for version 5 - when I called the function, I told it I wanted 21 days, not numDays days...so the calendar printed with 21 days no matter what I entered!!!

Sixth – Showed that we have the power to pass in different values for the # of days…we created a pattern where each subsequent month had 7 more days than the previous one.



<u>Functions that return something…like math functions</u>

Functions from math return things think of $f(x) = x^2$, so if the input is x = 5, the output is 25.

In programming functions are more flexible…

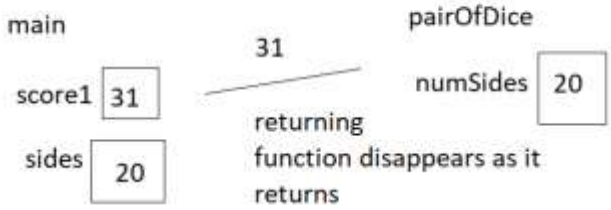
They don't have to take anything in and don't have to return anything. (main function)

They don't have to take anything in, but can still return something. (first pairOfDice function)

They don't have to return something (like the oneMonth function).

Or, they can do both…

Here we use different sided dice:



pairOfDice function – it's task is to return the sum of a pair of dice being rolled..

A function that returns something should **NOT** be called on a line by itself!!!

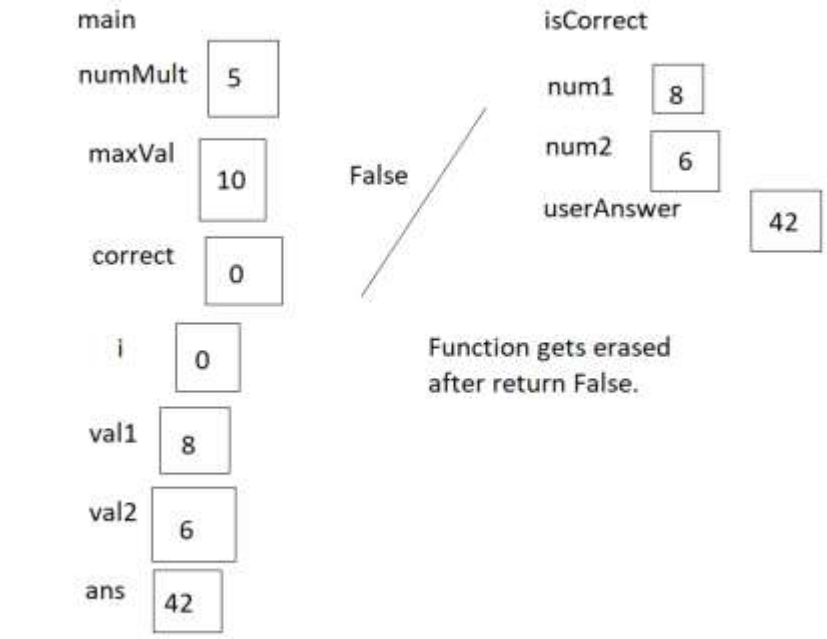
It needs to be called in a larger expression, so either what the function returns gets printed out, or it gets stored in a variable, or it gets used in a larger calculation…

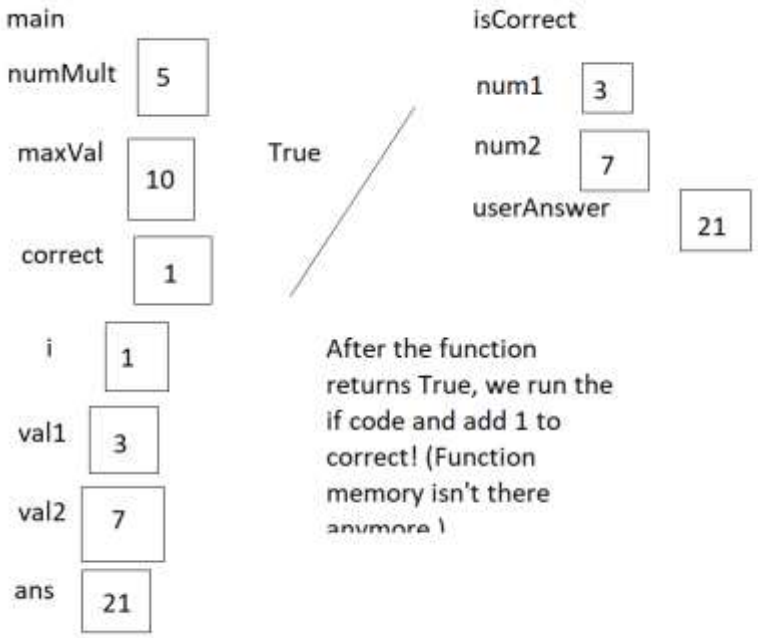
Game to write with functions

User chooses # of multiplication problems

Max input for each problem…

WE play the game…see how many the user gets correct


main
numMult
5
maxVal
10
correct
0
i
0
val1
8
val2
6
ans
42
False
isCorrect
num1
8
num2
6
userAnswer
42
Function gets erased
after return False.


main
numMult
5
maxVal
10
correct
1
i
1
val1
3
val2
7
ans
21
True
isCorrect
num1
3
num2
7
userAnswer
21
After the function
returns True, we run the
if code and add 1 to
correct! (Function
memory isn't there
anymore.)