We are now going to start improving on these strategies.

Each time we will pick the worst of the two and try to *leapfrog* over the other one to become the better of the two.

We want to make the decision that is returned *variable.*

So we will start by defining a *variable* –

By selecting the return instruction, pressing shift enter to get a new code prompt and then pressing v for variable

Now we must specify a name for the variable and we will call it ‘decision’ and then an *initial*  value which is the value of the variable unless it is subsequently changed.

So we will make that Action.stand – and enter

Then tab to the field in the return instruction, and edit it to return the value of the variable named decision.

We will start by emulating the strategy of the dealer – who will draw *if* the hand total is less than 17, but will otherwise stand.

So we will start by inserting an ‘if’ instruction … so, insert a new code prompt and type I

Now we must specify the condition for when we will do something – in this case that the *player’s* hand total is less than 17.

Now the player is being passed into the function as a parameter – given the name ‘p’

And if we type p dot we can see that there it has a value named handTotal

So we can write

If p dot handTotal

It is warning us that this isn’t yet a valid condition because handTotal is a number and we need something that is either true or false

But if we continue to write ‘is less than 17’ then we have a valid condition and we can hit Enter to go *into* the if instruction

Now we specify what we want done if, or when, that condition is true – which is to set the decision to Action draw

So s for set decision tab Action.draw

So – reading out the code - we’ve said that the initial – or we might say ‘default’ decision is stand except that if the player’s hand total is less than 17 then draw

Having make these changes, run the program.