PairProgramming_If_else

2024-07-24

if-else

Pair programming exercise for DSE5002, on if and else operations

when the test condition in an if statement is met, then the following code block is run. Otherwise, not

if(condition) { …put some commands here }

Suppose if x is less than 10, then we want to set y=0.125*x and z to x squared

This looks like

```
y=0
z=0
x=5


if(x>=10)# changed to x>=10
{
  y=0.125*x
  z=x^2
}
#cat is a print function I am using here, the \n is a linefeed character
# to advance to the next line in the printing

cat("Y is ",y ,"\n")
```

```
## Y is  0
```

```
cat("Z is ",z)
```

```
## Z is  0
```

Action Required: alter my code so that Y and Z are not altered

#If-else

Suppose that when x<10, we want this calculation to run as above, but for other x values, we want y=0.25*x, and z=square root of x

We use an else statement

```
y=0
z=0
x=5


if(x<10)
{
  y=0.125*x
  z=x^2

}else
{
  y=0.25*x
  z=x^(0.5)
}
#cat is a print function I am using here, the \n is a linefeed character
# to advance to the next line in the printing

cat(" When 'x'<10, if 'x'= ",x,"\n")
```

```
##  When 'x'<10, if 'x'=  5
```

```
cat("Y is ",y ,"\n")
```

```
## Y is  0.625
```

```
cat("Z is ",z,"\n")
```

```
## Z is  25
```

```
y=0
z=0
x=10


if(x<10)
{
  y=0.125*x
  z=x^2

}else
{
  y=0.25*x
  z=x^(0.5)
}
#cat is a print function I am using here, the \n is a linefeed character
# to advance to the next line in the printing

cat("when 'x' >= 10",x,"\n")
```

```
## when 'x' >= 10 10
```

```
cat("Y is ",y ,"\n")
```

```
## Y is  2.5
```

```
cat("Z is ",z)
```

```
## Z is  3.162278
```

Action Required:

Verify that this code works for a couple of different values of x

Note

Each if statement can only have one else

We can put if statements inside elses to allow for more possible options

```
x=18
print("change == to !=1")
```

```
## [1] "change == to !=1"
```

```
print(" %% will equal always equal remainder 1 when diveded by 2 and odd")
```

```
## [1] " %% will equal always equal remainder 1 when diveded by 2 and odd"
```

```
print(" can also change to %/% where remainder is always 0 if even, but %% is easiest to use for less complex statement")
```

```
## [1] " can also change to %/% where remainder is always 0 if even, but %% is easiest t
o use for less complex statement"
```

```
if(x<0)
{
  cat("Negative X value\n")
}else
{if(x%%2!=1)
  {
    cat("X is even\n")
  }
  else
  {
    cat("X is odd\n")
  }
}
```

```
## X is even
```

Alter this code and verify that it works

Compound test conditions

Using AND (&) and OR(|)

to decide what to do handle this decision

“I walk back the Starbucks some mornings, and while I like their coffee, the service is slow and I don’t like to wait. So I’ll stop for coffee there are 2 or less people in line. But if they have scones in stock, I’ll stop if there are 4 or less people in line”

Set up variables

people_in_line- which is an integer scones_in_stock-which is a binary or logical variable

What test condition would you need to figure out if I will stop at starbucks?

Set up an if statement that prints out the decision

```
people_in_line=3
scones_in_stock=FALSE

if((people_in_line <= 2)|(people_in_line <= 4 && scones_in_stock))
{
  cat("I'll stop by Starbucks\n")
}else
{
    cat("I won't stop by Starbucks\n")
}
```

```
## I won't stop by Starbucks
```

If-else assignment statements

R has the ability to carry out assignments in an if-else operator

we send in a condition and two possible assignments, the first for TRUE, the second for FALSE

it might look like this

```
x=9
y=ifelse(x<10,0.125*x,0.25*x)
cat("verifies 1st condition! y=",y,"\n")
```

```
## verifies 1st condition! y= 1.125
```

```
x=y*10
z=ifelse(x<10,0.125*x,0.25*x)
cat("verifies 2nd condition! z=",z)
```

```
## verifies 2nd condition! z= 2.8125
```

Action: Verify that this behaves as expected when x is changed

This is not a structure I use much, it does save some time. I tend to use the less sophisticated approach shown above.