

البرامج الإحصائية/2/

/السنة الرابعة-إحصاء رياضي/

المحاضرة السادسة والسابعة

التتابع والحلقات التكرارية

THE FUNCTIONS AND LOOPS

الدكتورة: فاطمة عبد الرحمن شلاف

للعام الدراسي 2019-2020

١- التتابع Functions

التتابع: هو مجموعة من التعليمات البرمجية تحمل اسماً محدداً وتؤدي مهمة معينة. الشكل العام للتتابع هو:

```
>function name<-function(arguments)
{
Statements
}
```

يتم استدعاء التابع بكتابة اسم التابع وبين قوسين تمرير وسطاء التابع.
ملاحظة: قد يكون التابع بدون وسطاء

مثال (1)

شكل تابعا يكتب العبارات التالية

```
Syrian Arab republic
University Of Albaath
Faculty of Science
```

الحل:

```
-----
> ff<-function()
{
print("Syrian Arab Republic")
print("University of Albaath ")
print("Faculty of Science")
}
-----
```

نتيجة التطبيق

```
> ff()
[1] "Syrian Arab Republic"
[1] "University of Albaath "
[1] "Faculty of Science"
-----
```

مثال (2)

شكل تابعا يقوم بحساب مجموع ثلاثة أعداد ومتوسطهم ويطبع النتائج.

الحل:

```
-----
> fs<-function(a,b,c)
+ {
+ s=a+b+c
+ m=s/3
+ print("the sum is")
+ print(s)
-----
```

```
+ print("the average is")
+ print(m)
+ }
```

استدعاء التابع بقيم عددية مباشرة

```
> fs(2,1,3)
[1] "the sum is"
[1] 6
[1] "the average is"
[1] 2
```

استدعاء التابع بمتحولات

```
> x=5
> y=6
> f=4
> fs(x,y,f)
[1] "the sum is"
[1] 15
[1] "the average is"
[1] 5
```

٢- اداة التحكم if
الشكل العام(1):

```
If(condition)
{
Statements
}
```

مثال(3)

شكل تابعاً بحيث يطبع كلمة positive إذا كانت قيمة العدد موجب.

```
>x<-5
>if(x>0)
{
Print("positive"
}
[1] "positive"
```

مثال(4)

اكتب الكود اللازم لحساب قيمة y المعرقة كمايلي

$$Y=x^2/k$$

الحل:

بما أن k لا يمكن أن تساوي الصفر لذلك لا بد من وضع شرط قبل تنفيذ حساب قيمة y .

نكتب التابع التالي لتنفيذ المطلوب

```
-----  
> ff<-function(x,k)  
{  
if(k==0)  
{  
print("Error")  
}  
y=x^2/k  
y  
}  
-----
```

التطبيق:

```
-----  
> ff(6,4)  
[1] 9  
> ff(7,0)  
[1] "Error"  
[1] Inf  
-----
```

الشكل العام(2)

```
if(condition)  
{  
Statements  
}else  
{  
Statement  
}
```

مثال (5)

شكل تابعاً بحيث يطبع كلمة negative إذا كانت قيمة العدد سالبة، و positive إذا كانت قيمة العدد موجبة.

```
-----  
>x=-9  
>if(x<0)  
{  
Print("negative")  
}else  
{  
Print("positive")  
}  
[1] "negative"  
-----
```

مثال (6)

أكتب الكود اللازم لحساب قيمة y المعرفة كمايلي:

$$y = \begin{cases} x + 3 & \text{if } x \geq 5 \\ x^3 & \text{in other cases} \end{cases}$$

الحل:

نشكل تابع لحساب قيمة y المعرفة أعلاه

```
-----  
> ff1<-function(x)  
{  
if(x>=5)  
{  
y=x+3  
}else  
{  
y=x^3  
}  
print(y)  
}  
-----
```

التطبيق:

```
-----  
> ff1(5)  
[1] 8  
> ff1(-3)  
[1] -27  
-----
```

مثال(7)

اكتب الكود اللازم لحساب قيمة y المعرفة كمايلي:

$$Y = \begin{cases} x/m & \text{if } x > 4 \\ x^3 + 5 & \text{in other cases} \end{cases}$$

الحل:

نشكل تابعاً لحساب قيمة y المعرفة أعلاه مع الأخذ بعين الاعتبار الا تساوي m الى الصفر.

```
-----  
> ff2<-function(x,m)  
{  
if(x>4)  
{  
if(m==0)  
{  
print("Error m=0")  
stop  
}  
y=x/m  
}else  
{  
y=x^3+5  
}  
print(y)  
}  
-----
```

استدعاء التابع والوقوع في الحالة الأولى

```
> ff2(5,3)  
[1] 1.666667  
-----
```

استدعاء التابع والوقوع في الحالة الثانية

```
> ff2(4,3)
```

```
[1] 69
```

استدعاء التابع والوقوع في الحالة الأولى مع وجود خطأ

```
> ff2(6,0)
```

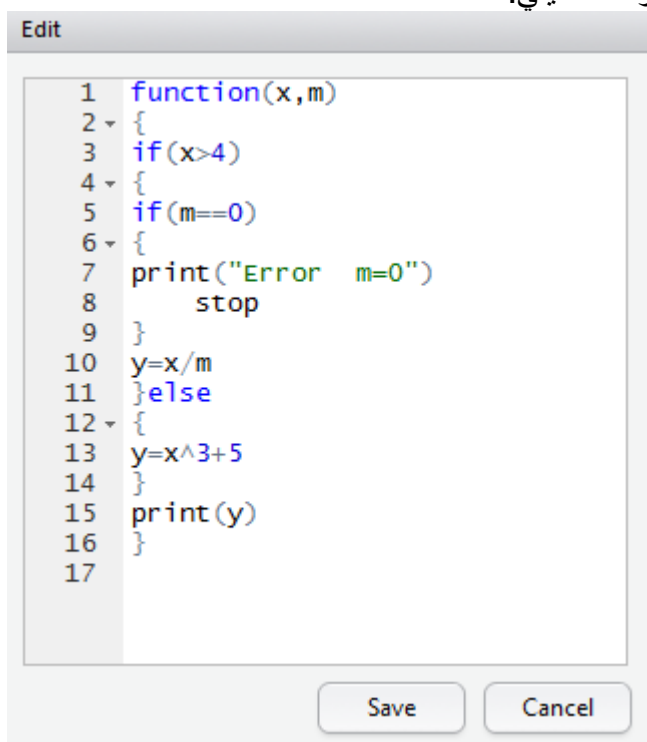
```
[1] "Error m=0"
```

```
[1] Inf
```

في حال كان هناك خطأ في كتابة التابع ونريد التصحيح نكتب:

```
> ff2<-edit(ff2)
```

فتخرج معنا لوحة كمايلي:



```
Edit
1 function(x,m)
2 {
3   if(x>4)
4   {
5     if(m==0)
6     {
7       print("Error m=0")
8       stop
9     }
10  y=x/m
11  }else
12  {
13    y=x^3+5
14  }
15  print(y)
16  }
17
```

الشكل (1) يبين نافذة تصحيح التابع

نصحح المطلوب ونضغط على مفتاح save

الشكل العام (3)

```
If(condition)
```

```
{
```

```
Statements
```

```
}else if(condition)
```

```
{
```

Statements

```
}else if(condition)
```

```
.....
```

مثال(8)

```
-----  
>rr<- function(x)  
{
```

```
  if(x<0)
```

```
  {
```

```
    print("negative")
```

```
  }
```

```
  else if(x>0)
```

```
  {
```

```
    print("positive")
```

```
  }else
```

```
  {
```

```
    print("zero")
```

```
  }
```

```
}
```

```
> rr(-1)
```

```
[1] "negative"
```

مثال(9)

اكتب الكود اللازم لحساب قيمة y المعرفة كمايلي:

$$Y = \begin{cases} x^2 + 1 & \text{if } 0 \leq x \leq 5 \\ x - 3 & \text{if } 5 < x \leq 20 \\ x^3/3 & \text{in other cases} \end{cases}$$

الحل:

```
-----  
> ff3<-function(x)
```

```
{
```

```
  if(x>=0 & x<=5)
```

```
  {
```

```
    y=x^2+1
```

```
  }else if(x>5 & x<=20)
```

```
  {
```

```
    y=x-3
```

```
  }else
```

```
  {
```



```
y=x^3/3
}
print ("the value of y")
print(y)
}
```

استدعاء التابع والوقوع في الحالة الأولى

```
> ff3(4)
[1] "the value of y"
[1] 17
```

استدعاء التابع والوقوع في الحالة الثانية

```
> ff3(6)
[1] "the value of y"
[1] 3
```

استدعاء التابع والوقوع في الحالة الثالثة

```
> ff3(-3)
[1] "the value of y"
[1] -9
```

٣- الأداة ifelse

الشكل العام:

```
ifelse (condition,a,b)
```

إذا كان الشرط صحيح تنفذ العملية a وإن كان غير صحيح تنفذ العملية b

مثال(10)

```
> x
[1] 1 -6 0 5 11 -2
> ifelse(x>0,"positive","negative")
[1] "positive" "negative" "negative" "positive" "positive" "negative"
```

مثال(11)

شكل تابعاً بحيث يختبر العدد ان كان فردياً أو زوجياً ويطبع odd للفردى و even للزوجى مستخدماً تعليمة ifelse.

الحل:

```
> fss<-function(x)
{
ifelse(x%%2==1,"odd","even")
}
```

```
-----  
> x=12  
> y=7  
> fss(x)  
[1] "even"
```

```
-----  
> fss(7)  
[1] "odd"
```

```
-----  
> fss(-3)  
[1] "odd"
```

```
-----  
> fss(10)  
[1] "even"
```

٤- الحلقة for
الشكل العام:

```
for(i in start:end)  
{  
  Statements  
}
```

مثال (12)

اطبع مربع الأعداد التالية (5,12,13) مستخدماً أمر الحلقة.

```
-----  
> x<-c ( 5 12 13)  
> for (n in x)  
{  
  print(n^2)  
}  
[1] 25  
[1] 144  
[1] 169
```

مثال (13)

لدينا المصفوفة التالية:

```
-----  
> x  
      [,1] [,2] [,3]  
[1,]  1   4   7
```

```
[2,] 2 5 8
[3,] 3 6 9
```

المطلوب: استخدام الحلقات لـ:
١. طباعة عناصر المصفوفة
٢. طباعة عناصر القطر الرئيسي
أولاً:

```
> for(i in 1:3)
  for(j in 1:3)
  {
  print(x[i,j])
  }
[1] 1
[1] 4
[1] 7
[1] 2
[1] 5
[1] 8
[1] 3
[1] 6
[1] 9
```

لاحظ أنه يتم الطباعة سطراً سطراً
ثانياً:

```
> for(i in 1:3)
  {
  print(x[i,i])
  }
[1] 1
[1] 5
[1] 9
```

مثال(14)
لتكن لدينا المصفوفة التالية:

```
> y<-matrix(c(1,2,3,4,5,6,7,8,9),ncol=3)
> y
     [,1] [,2] [,3]
```

```
[1,] 1 4 7
[2,] 2 5 8
[3,] 3 6 9
```

طباعة العناصر التي تقع فوق القطر الرئيسي (طبعاً هي العناصر التي يكون فيها دليل السطر أقل من دليل العمود $i < j$)

```
> for(i in 1:3)
  for(j in 1:3)
  {
  if(i<j)
  {
  print(y[i,j])
  }
  }
[1] 4
[1] 7
[1] 8
```

طباعة العناصر التي تقع تحت القطر الرئيسي (هي العناصر التي يكون فيها دليل السطر أكبر من دليل العمود $i > j$)

```
> for(i in 1:3)
  for(j in 1:3)
  {
  if(i>j)
  {
  print(y[i,j])
  }
  }
[1] 2
[1] 3
[1] 6
```

طباعة العناصر التي تقع على القطر الرئيسي (هي العناصر التي يكون فيها دليل السطر يساوي دليل العمود $j=i$)

```
> for(i in 1:3)
  for(j in 1:3)
  {
```

```

if(i==j)
{
print(y[i,j])
}
}
[1] 1
[1] 5
[1] 9

```

أو يمكن طباعتها كمايلي:

```

> for(i in 1:3)
+ {
+ print(y[i,i])
+ }
[1] 1
[1] 5
[1] 9

```

طباعة العناصر التي تقع فوق القطر الثانوي (هي العناصر التي يكون فيها دليل السطر مجموعاً مع دليل العمود تكون أصغر من رتبة المصفوفة $z+i \leq 3$)

```

> for(i in 1:3)
for(j in 1:3)
{
if(i+j<=3)
{
print(y[i,j])
}
}
[1] 1
[1] 4
[1] 2

```

طباعة العناصر التي تقع تحت القطر الثانوي (هي العناصر التي يكون فيها دليل السطر مجموعاً مع دليل العمود أكبر من رتبة المصفوفة مضاف لها 2 أي $(i+j) \geq 5$)

```

for(i in 1:3)
for(j in 1:3)
{

```

```

if(i+j>=5)
{
print(y[i,j])
}
}
[1] 8
[1] 6
[1] 9

```

طباعة العناصر التي تقع على القطر الثانوي (هي العناصر التي يكون فيها دليل السطر مضافاً إليه دليل العمود يساوي رتبة المصفوفة مضاف إليه واحد أي $(i+j=4)$)

```

for(i in 1:3)
for(j in 1:3)
{
if(i+j==4)
{
print(y[i,j])
}
}
}
[1] 7
[1] 5
[1] 3

```

٥- الحلقة التكرارية while
الشكل العام:

```

While(condition)
{
Statements
}

```

مثال (15)

اكتب الكود اللازم لإدخال مجموعة من الأعداد وحساب مجموعهم والتوقف عن العملية إذا أصبح المجموع أكبر من 10 .

```

> x=0
while(x<=10)
{
print(x)
x=x+2
}

```

```
+ }
[1] 0
[1] 3
[1] 6
[1] 9
```

أمثلة (16)

أولاً:

```
> i=0
> while(i<5)
{
print(i^3)
i=i+1
}
[1] 0
[1] 1
[1] 8
[1] 27
[1] 64
```

ثانياً:

```
> i <- 1
while (i <= 10) i <- i+4
> i
[1] 13
```

ثالثاً:

مشابه تماماً للكود في الأعلى

```
> i <- 1
```

مثال (17)

أكتب الكود اللازم لطباعة العناصر التي تقع فوق القطر الرئيسي في المصفوفة التالية مستخدماً حلقات while .

```
[,1] [,2] [,3]
[1,] 1 4 7
[2,] 2 5 8
[3,] 3 6 9
```

الحل:

```

> ff<-function(x)
{
i=0
while(i<=3)
{
i=i+1
j=0
while(j<=3)
{
if(i<j)
{
print(x[i,j])
}
j=j+1
}
}
}
> ff(x)
[1] 4
[1] 7
[1] 8

```

إذا أردنا طباعة العناصر التي تقع تحت القطر الرئيسي نكتب نفس الكود ولكن الشرط نغيره إلى: $if(i>j)$ إذا أردنا طباعة العناصر التي تقع على القطر الرئيسي نكتب نفس الكود ولكن الشرط نغيره إلى: $if(i==j)$ وبنفس الطريقة كما في حلقات for نغير الشرط لنحصل على العناصر تحت وفوق وعلى القطر الثانوي.

٦- التعليمة break

سوف نشرح التعليمة من خلال المثال التالي الذي يستمر بطباعة العناصر المحصورة بين 1 و 100 إلى أن نصادف رقم يكون باقي قسمته على 3 وعلى 4 يساوي الصفر.

```

> for(i in 1:100)
{
if(i%%3==0 & i%%4==0)break
print(i)
}
[1] 1
[1] 2
[1] 3

```



```
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
[1] 11
```

٧- التعليمه next

سوف نشرح التعليمه أيضاً من خلال المثال التالي الذي يطبع العناصر بين 1 و الـ 20 والتي يكون باقي قسمتها على الرقم 4 يساوي 3:

```
> for(i in 1:20)
{
if(i%%4!=3)next
print(i)
}
[1] 3
[1] 7
[1] 11
[1] 15
[1] 19
```

ونفس النتيجة بكتابة:

```
for(i in 1:20)
+ {
+ if(i%%4==3)
+ print(i)
+ }
[1] 3
[1] 7
[1] 11
[1] 15
[1] 19
```

٨- التعليمه repeat الشكل العام:

```
Repeat
{
Statements
}
```

مثال(18)

اطبع الأعداد الزوجية المحصورة بين 1 و 10
الحل:

```
-----  
> i=1  
> repeat  
+ {  
+ if(i%%2==0)  
+ print(i)  
+ i=i+1  
+ if(i>10)break  
+ }  
[1] 2  
[1] 4  
[1] 6  
[1] 8  
[1] 10  
-----
```