

In [1]:

```
#print -> user friendly (or) easily understandable  
#paste -> sometimes user cannot understand
```

DATA FRAMES:

In [15]:

```
df=data.frame(emp_id=c(1,2,3),  
              emp_name=c("Nanthiesh","Demon","NarutoUzumaki"),  
              emp_date=c("2004-07-27", "1001-01-01", "1001-03-01"),  
              gender=factor(c("M","M","F")))
```

In [3]:

df

emp_id	emp_name	emp_date	gender
1	Nanthiesh	2004-07-27	M
2	Demon	1001-01-01	M
3	NarutoUzumaki	1001-03-01	F

In [4]:

```
# sorting the data with joining date  
df[with(df,order(c(emp_date))),]
```

	emp_id	emp_name	emp_date	gender
2	2	Demon	1001-01-01	M
3	3	NarutoUzumaki	1001-03-01	F
1	1	Nanthiesh	2004-07-27	M

In [5]:

```
# sorting the data with employee name  
df[with(df,order(c(emp_name))),]
```

	emp_id	emp_name	emp_date	gender
2	2	Demon	1001-01-01	M
1	1	Nanthiesh	2004-07-27	M
3	3	NarutoUzumaki	1001-03-01	F

In [24]:

```
# adding new column in the existing data frame  
df$dept=c("CSE","PYTHON","MATHS")
```

In [25]:

df

emp_id	emp_name	emp_date	gender	dept
1	Nanthiesh	2004-07-27	M	CSE
2	Demon	1001-01-01	M	PYTHON
3	NarutoUzumaki	1001-03-01	F	MATHS

In [20]:

```
new_df=data.frame(emp_id=c(5),  
                  emp_name=c("NANTHIESH"),  
                  emp_date=c("2006-07-27"),  
                  gender=factor(c("M")),  
                  dept=c("IT"))
```

In [26]:

new_df

emp_id	emp_name	emp_date	gender	dept
5	NANTHIESH	2006-07-27	M	IT

In [27]:

```
# binding the new data frame with the old data frame  
rbind(df,new_df)
```

emp_id	emp_name	emp_date	gender	dept
1	Nanthiesh	2004-07-27	M	CSE
2	Demon	1001-01-01	M	PYTHON
3	NarutoUzumaki	1001-03-01	F	MATHS
5	NANTHIESH	2006-07-27	M	IT

In [30]:

```
# removing coulmn or drop columns  
df=subset(df, select=-c(dept))
```

In [31]:

df

emp_id	emp_name	emp_date	gender
1	Nanthiesh	2004-07-27	M
2	Demon	1001-01-01	M
3	NarutoUzumaki	1001-03-01	F

In [34]:

```
# removing the row (or) dropping a row
# before removing
df[c(2),]
# after removing
df[-c(2),]
```

	emp_id	emp_name	emp_date	gender
2	2	Demon	1001-01-01	M

	emp_id	emp_name	emp_date	gender
1	1	Nanthiesh	2004-07-27	M
3	3	NarutoUzumaki	1001-03-01	F

JOINS:

In [41]:

```
df8=data.frame(StudentId=c(101,102,103,104,105,106,107),
                 Product=c("Hindi","English","Tamil","Maths","Biology","Social Science","Zoolog
```

In [42]:

df8

StudentId	Product
101	Hindi
102	English
103	Tamil
104	Maths
105	Biology
106	Social Science
107	Zoology

In [43]:

```
df9=data.frame(StudentId=c(102,103,104,105,106),
                 State=c("Chennai","Telungana","Mysore","Pune","Delhi"))
```

In [44]:

```
df9
```

StudentId	State
102	Chennai
103	Telungana
104	Mysore
105	Pune
106	Delhi

merge(x,y,by,all)

Left Join:

In [46]:

```
df5=merge(x=df8,y=df9, by="StudentId",all.x=TRUE)  
df5
```

StudentId	Product	State
101	Hindi	NA
102	English	Chennai
103	Tamil	Telungana
104	Maths	Mysore
105	Biology	Pune
106	Social Science	Delhi
107	Zoology	NA

Right Join:

In [48]:

```
df4=merge(x=df8,y=df9,by="StudentId",all.y=TRUE)  
df4
```

StudentId	Product	State
102	English	Chennai
103	Tamil	Telungana
104	Maths	Mysore
105	Biology	Pune
106	Social Science	Delhi

Natural Join:

In [50]:

```
df5=merge(x=df8,y=df9,by="StudentId",all=FALSE)
df5
```

StudentId	Product	State
102	English	Chennai
103	Tamil	Telungana
104	Maths	Mysore
105	Biology	Pune
106	Social Science	Delhi

Full Outer Join:

In [51]:

```
df6=merge(x=df8,y=df9,by="StudentId",all=TRUE)
df6
```

StudentId	Product	State
101	Hindi	NA
102	English	Chennai
103	Tamil	Telungana
104	Maths	Mysore
105	Biology	Pune
106	Social Science	Delhi
107	Zoology	NA

Looping:

if else syntax:

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

In [52]:

```
a=10
if(a%2==0)
{
    print("Even")
} else
{
    print("Odd")
}
```

[1] "Even"

In [54]:

```
# break statement:
l=list(1,2,3,4,5)
for(x in l)
{
    if(x==2)
    {
        break
    }
    print(x)
}
```

[1] 1

In [55]:

```
# next statement:
l=list(1,2,3,4,5)
for(x in l)
{
    if(x==2)
    {
        next
    }
    print(x)
}
```

[1] 1

[1] 3

[1] 4

[1] 5

In [57]:

```
# using sequence:
l=list(1,2,3,4,5)
for(x in seq(1,10))
{
    if(x==2)
    {
        next
    }
    print(x)
}
```

```
[1] 1
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
```

ARMSTRONG NUMBER:

Given number: 153

$1^3 + 5^3 + 3^3$

$1+125+27 = 153$

```
while(num>0)
{
    digit=num%%10
    sum=sum+ (digit^3)
    num=floor(num/10)
}
```

In [69]:

```
num=as.integer(readline("Enter your value: "))
sum=0
temp=num
while(num>0)
{
    digit=num%%10
    sum=sum+ (digit^3)
    num=floor(num/10)
}
if(temp==sum)
{
    print("The Given Number is a Armstrong Number")
} else
{
    print("The Given Number is not a Armstong Number")
}
```

Enter your value: 153

```
[1] "The Given Number is a Armstrong Number"
```

SUM OF ALL DIGITS:

In [71]:

```
num=as.integer(readline("Enter your value: "))
sum=0
temp=num
while(num>0)
{
    digit=num%%10
    sum=sum+digit
    num=floor(num/10)
}
print(paste("The Sum Value is: ",sum))
```

Enter your value: 15

[1] "The Sum Value is: 6"

REVERSE OF A NUMBER:

In [74]:

```
num=as.integer(readline("Enter your value: "))
rev=0
temp=num
while(num>0)
{
    digit=num%%10
    rev=rev*10+digit
    num=floor(num/10)
}
print(paste("The Reverse Value of ",temp, "is",rev))
```

Enter your value: 123

[1] "The Reverse Value of 123 is 321"

PALINDROME:

In [75]:

```
num=as.integer(readline("Enter your value: "))
rev=0
temp=num
while(num>0)
{
    digit=num%%10
    rev=rev*10+digit
    num=floor(num/10)
}
if(temp==rev)
{
    print("The Given Number is a Palindrome")
} else
{
    print("The Given Number is not a Palindrome")
}
```

Enter your value: 123

[1] "The Given Number is not a Palindrome"

In [76]:

```
num=as.integer(readline("Enter your value: "))
rev=0
temp=num
while(num>0)
{
    digit=num%%10
    rev=rev*10+digit
    num=floor(num/10)
}
if(temp==rev)
{
    print("The Given Number is a Palindrome")
} else
{
    print("The Given Number is not a Palindrome")
}
```

Enter your value: 151

[1] "The Given Number is a Palindrome"

In []: