

CLASSES & OBJECT:

In [1]:

```
# creation of class:
employee1=list(name="xyz", dept="CSE")
class(employee1) # class creation
employee1 # object creation
```

'list'

\$name

'xyz'

\$dept

'CSE'

In [2]:

```
employee1$dept
```

'CSE'

Create a class called student the student name unique id and print only the term 2 course:

In [5]:

```
student=list(name=c("Nanthiesh", "Devil", "Table"), uniqueid=c("e0121015", "e0121013", "e0121056"), course=c(1,2,2))
class(student)
student
```

'list'

\$name

'Nanthiesh' 'Devil' 'Table'

\$uniqueid

'e0121015' 'e0121013' 'e0121056'

\$course

1 2 2

In [6]:

```
student
```

1 2 2

Debugging:

Debugging is a process of cleaning a program code from bugs to run it successfully while writing the codes some mistakes or problems automatically appears after the completion of the code, so fixing it takes a lot of time and after multiple level of calls so we can do the debugging in R using warning messages

Various Debugging Functions:

- * Traceback
- * Browser
- * Recover

options error = traceback

In []:

```

exponent <- function (num, pow){
  if (pow== 0){
    return(1)
  }
  if (pow==1){
    return (num + "Morfmaert")
  }
  if (pow%% 2 == 1){
    return(num exponent (num, pow - 1))
  } else {
    temp <- exponent (num, floor (pow / 2))
    return(temp temp)
  }
}
options error traceback
exponent (10, 10)

```

In []:

Regular Expression:

Regular expression is a powerful tool for string manipulation

- * grepl() -> returns the boolean value
- * grep() -> returns the index value
- * str_extract_all()

(grepl) Function:

In [10]:

```

words = c("python", "java", "Python", "C")
grepl(pattern="Python", x=words,ignore.case=TRUE)

```

TRUE FALSE TRUE FALSE

In [12]:

```

words = c("python", "java", "Python", "C")
grep(pattern="Python", x=words,ignore.case=TRUE) # grep will give only the index value which matches the conditon

```

1 3

In [14]:

```

words = c("I love python", "java", "I hate Python", "C")
grep(pattern="Python",words, ignore.case=TRUE, value=TRUE) # value is used to print the whole statement

```

'I love python' 'I hate Python'

Extract the phone number from the given paragraph:

In [1]:

```
install.packages("stringr")
```

also installing the dependencies 'cli', 'glue', 'lifecycle', 'rlang', 'stringi', 'vctrs'

There are binary versions available but the source versions are later:

	binary	source	needs_compilation
cli	2.5.0	3.4.1	TRUE
glue	1.4.2	1.6.2	TRUE
lifecycle	1.0.0	1.0.3	FALSE
rlang	0.4.11	1.0.6	TRUE
stringi	1.6.1	1.7.8	TRUE
vctrs	0.3.8	0.5.1	TRUE
stringr	1.4.0	1.5.0	FALSE

Binaries will be installed

package 'cli' successfully unpacked and MD5 sums checked
 package 'glue' successfully unpacked and MD5 sums checked
 package 'rlang' successfully unpacked and MD5 sums checked
 package 'stringi' successfully unpacked and MD5 sums checked
 package 'vctrs' successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\nanth\AppData\Local\Temp\RtmpiaFFh2\downloaded_packages

installing the source packages 'lifecycle', 'stringr'

Warning message in install.packages("stringr"):

"installation of package 'lifecycle' had non-zero exit status"Warning message in install.packages("stringr"):

"installation of package 'stringr' had non-zero exit status"

```
[9] library(stringr)
x=c("My name is Nanthiesh and my phone number is 7150563929 and my id is E0121015")
str_extract_all(x,"[6-9]{1}[0-9]{9}")

1. '7150563929'
```

```
[13] library(stringr)
x=c("My name is Nanthiesh and my phone number is 7150563929 and my id is E0121015 and the date is 27-07-1999")
phonenumber = str_extract_all(x,"[6-9]{1}[0-9]{9}")
date = str_extract_all(x,"[:digit:]{2}-[:digit:]{2}-[:digit:]{4}")
print(paste("phonenumber: ", phonenumber))
print(paste("date: ", date))

[1] "phonenumber: 7150563929"
[1] "date: 27-07-1999"
```

```
[15] library(stringr)
x=c("My name is Nanthiesh and my phone number is 7150563929 and my id is E0121015 and the date is 27-07-1999 and my mail id is nanthunanthies@gmail.com")
mail = str_extract_all(x,"[\\w]+@[\\w]+.[\\w]+")
print(paste("Mail Id: ",mail))

[1] "Mail Id: nanthunanthies@gmail.com"
```

In [3]:

```
library("stringr")
```

Error in library("stringr"): there is no package called 'stringr'

Traceback:

1. library("stringr")

extract date:

```
[:digit:]{2}-[:digit:]{2}-[:digit:]{4}
```

extract phone number:

```
[6-9]{1}[0-9]{9}
```

extract mail id:

```
[\\w]+@[\\w]+.[\\w]+
```

Write a R code to print the first two digit followed by the arbitrary character:

```
library(stringr)
x=c("Recent research has provided a wealth of insight about how dogs came to be domesticated by humans 9845752352 and the roles they played in Native American culture.
DNA studies on archaeological finds suggest that dogs may have been domesticated by humans as long as 40,000 years ago. When the first humans came to North
America from Eurasia, at least 12,000 years ago, domesticated dogs came with them. They appear to have been 12-07-2004 highly prized by early North American hunter-gatherers
and were their only animal companions for centuries, since there were no horses on nanthunanthiesh@gmail.com the continent until the 16th century.")
phonenumber = str_extract_all(x,"[6-9]{1}[0-9]{9}")
date = str_extract_all(x,"[:digit:]{2}-[:digit:]{2}-[:digit:]{4}")
mail = str_extract_all(x,"[\\w]+@[\\w]+\\.([\\w]+)")
print(paste("phonenumber: ", phonenumber))
print(paste("date: " , date))
print(paste("Mail Id: ",mail))

[1] "phonenumber: 9845752352"
[1] "date: 12-07-2004"
[1] "Mail Id: nanthunanthiesh@gmail.com"
```

```
[19] library(stringr)
a=list("12 abc is my address 123 is ward number ")
str_extract_all(a, "\\b\\d{2}\\b\\s[\\w]+")
```

```
1. '12 abc'
```

```
[21] library(stringr)
a=list("12 abc is my address 123 is ward number ")
str_extract_all(a, "w[\\w]+d") # start with w and end with d
```

```
1. 'ward'
```

To Replace a String or a Value:

- * sub -> replaces value at first only
- * gsub -> replaces values in all the places

In [21]:

```
df <- "Python is a collaborative project with many contributors-R and R is a high level language"
sub('R','R language',df)
```

```
'Python is a collaborative project with many contributors-R language and R is a high level language'
```

In [22]:

```
df <- "Python is a collaborative project with many contributors-R and R is a high level language"
gsub('R','R language',df) #(old,new,input)
```

```
'Python is a collaborative project with many contributors-R language and R language is a high level language'
```

grep() Function:

Print a word that contains the vowels a,e,i,o,u

In [23]:

```
s=c("Nanthiesh", "BAC", "XYZ", "code", "r")
grep(pattern="[aeiouAEIOU]",s,value=TRUE)
```

```
'Nanthiesh' 'BAC' 'code'
```