| Danuary 2008 Mon 05 12 19 26 Tue 06 13 20 27 Wed 07 14 21 28 Thu 01 08 15 22 29 Fri 02 09 16 23 30 Sat 03 10 17 24 31 Sun 04 11 18 25 |
|---|
| ⇒ val = 50 → cant be changed ⇒ var = 50 → can be changed |
| => Datatypes = String, Int, Double, Boolean |
| => Datatypes = String; Int, Double, Boolean => var name: String = "Yash" or val => print ("Yash") -> print valve |
| => println("Yash") print value followed by a new-line |
| => val age: String = ""+4.2+6+"!" valid, use any values to concatenate them, but they must be in between two strings |
| e) Initializing long: val i = 10L |
| val i3: Long = 10 val i3: Long = 10 >> Float type is also present, but it has lower precision. val pi = 3-14 F. |
| Type casting = i=10 Notes var q = i . to Long! |

| 1 | 13 | Decemb | |
|---|----------|---------------|--|
| | <u>U</u> | Saturday | |
| > | += | · - = · · * = | |



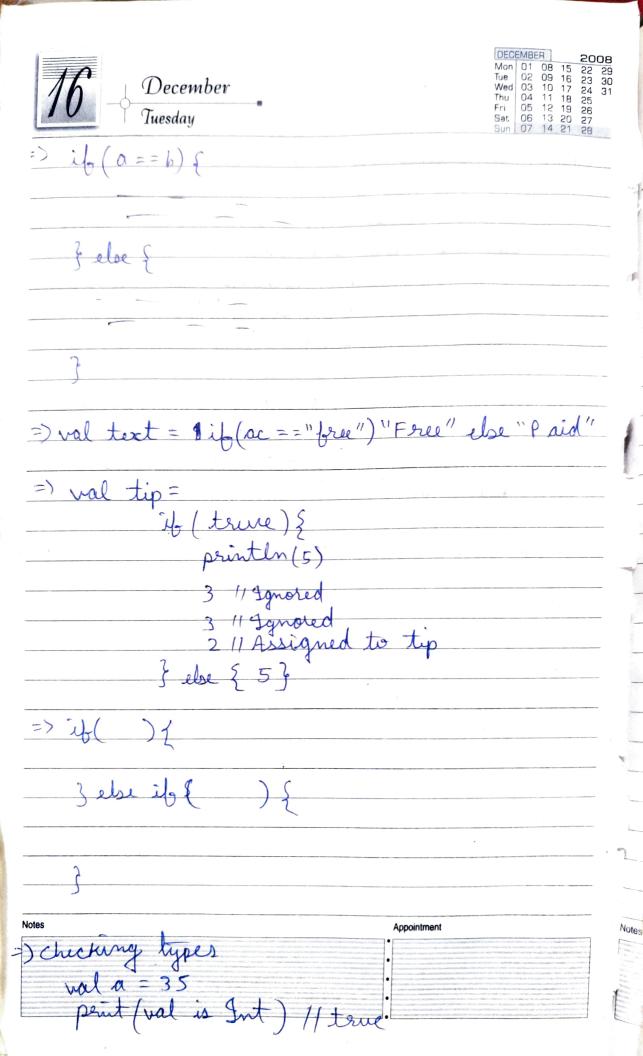
Notes

Appointment

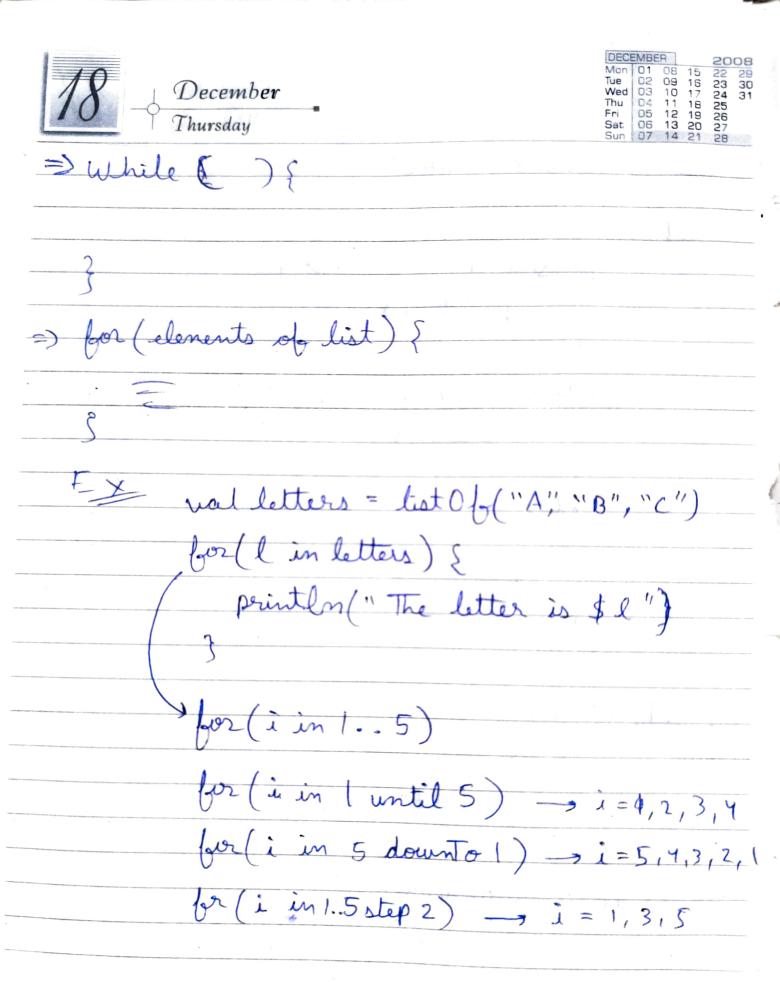
| Mon | December Monday |
|--------------------------|----------------------------|
| => Boolean functions | with texts |
| val sal = "Hello | . 17 |
| val tr = sal-s | tarts With ("Hel") // true |
| | ends With ("lo") // true |
| • | first() // H' |
| | last () // 'o' |
| | equals ("Hello") // true |
| | uppercase () // HELLO |
| | lowercase () // hello |
| | substring (2) //llo |
| fir = sal[0] | // H' |
| U | |
| =) & & , 11 , != , = = , | 7, <, >=, <= present too |
| | |
| | |
| | |
| | |
| | • |
| | |
| | |
| | |
| | |
| | |

Appointment

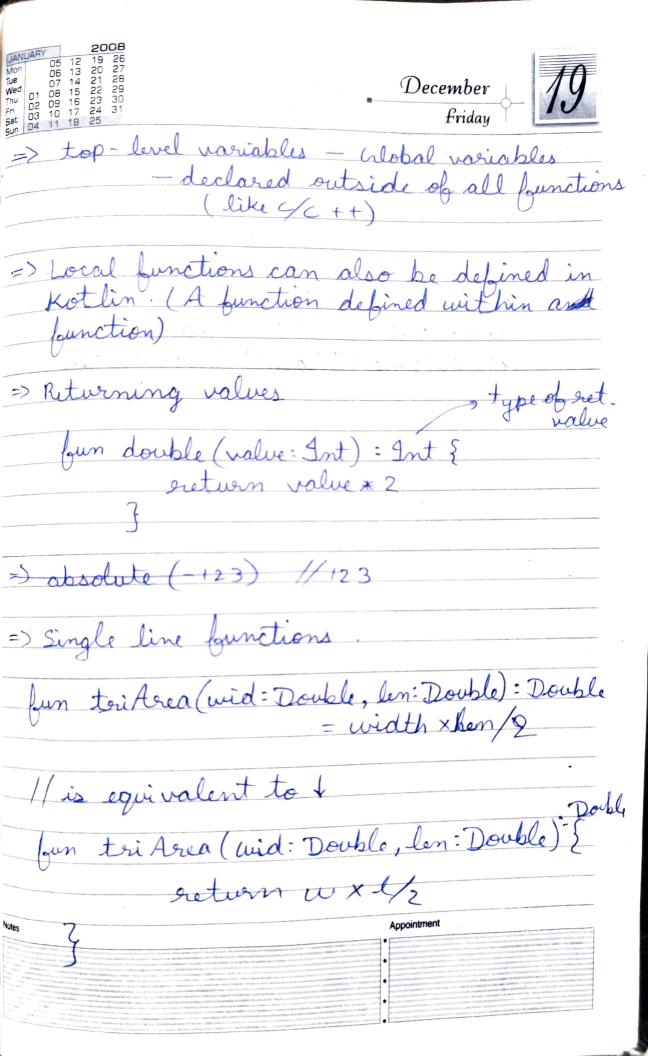
Notes



05 12 19 06 13 20 07 14 21 08 15 22 09 16 23 10 17 24 11 18 25 December Wednesday password = "ABC" when { passioned == " " -> { print(1) password. length < 7 -> { print (2) println ("OK") => when (num) { peint (" one") val a = when(mum) { "Hello" print (a) [l"Hello" =) to sheek ranges 2,3,5 -> { } 2.5 -> { 3



INCRES



Sun 07 14 21 => Named So default arguements fun cheer (now: String = "Hello", who: String = "World"); cheer ("Hi") // Hi World cheer ("Hi") // Hi World cheer (who = "Yash") // Hello Yash