

```
// 1. Perhatikan code berikut ini! Hasil dari console.log(angka) adalah
/*
  let angka = 12
  angka *= `2`
  angka++
  angka++
  angka += '1'
  console.log(angka)
*/
// a. "261"
// b. 261
// c. 1241
// d. "1241"
```

```
// 2. Apakah hasil console pada program di bawah ini?
/*
  let strNumber = '20abc'
  strNumber = Number(strNumber) // NaN
  strNumber = String(strNumber) // "NaN"
  console.log(strNumber) // "NaN"
  console.log(typeof strNumber) // String
*/
// a. 20 & Number
// b. NaN & String
// c. Null & String
// d. NaN & NaN
```

```
// 3. Berikut ini penamaan variable yang tepat sesuai dengan aturan Javascript, yaitu?
// a. let #myName
// b. var $myName
// c. const _myName
// d. Semua jawaban salah
```

```
// 4. Berikut ini yang merupakan tipe data primitive, kecuali?
// a. String
// b. Array
```

// c. Number
// d. Null

// 5. Dibawah ini manakah variable yang menghasilkan value undefined
// apabila variable tersebut di akses terlebih dahulu sebelum didefine?
// a. var
// b. let
// c. const
// d. Semua jawaban salah

// 6. Perhatikan code berikut! Hasil dari console di bawah ini apabila di jalankan di terminal adalah?

```
/*  
  const name = "Budi"  
  console.log("Nama saya ${name}")  
*/  
// a. Nama saya Budi  
// b. Nama saya "Budi"  
// c. TypeError: Assignment to constant variable.  
// d. Semua jawaban salah
```

// 7. Berikut ini, manakah pernyataan yang paling tepat?
// a. Variable yang dideclare menggunakan var, value nya tidak dapat diubah
// b. Variable yang dideclare menggunakan const, valuenya dapat diubah
// c. Variable yang dideclare menggunakan let, tidak memiliki aturan scope
// d. Variable yang dideclare menggunakan var, tidak bisa di duplicate

// 8. Berikut ini pernyataan yang tepat mengenai conditional statement adalah?
// a. Untuk menjalankan program secara berulang-ulang
// b. Untuk mengambil keputusan
// c. Untuk menampilkan data yang telah di proses
// d. Untuk menghentikan suatu baris code

// 9. Perhatikan code berikut. Hasil dari console.log berikut adalah?

```
/*
let point1 = 10
let point2 = 100
let result = point1 < point2
console.log(result)
*/
// a. 10
// b. 100
// c. true
// d. false
```

// 10. Bagaimanakah hasil dari program berikut apabila kita running?

```
/*
  let point = 5

  do{
    point++ // 6
    console.log("Point Bertambah") // 1x
    point = 0 // 6 ---> 0
  }while(point <= 5)

  while(point >= 5){
    console.log("Point Terus Bertambah")
  }
*/
// a. Looping do while dijalankan 1x dan looping while akan infinite loop
// b. Looping do while akan infinite loop dan looping while akan dijalankan 1x
// c. Looping do while akan infinite loop dan looping while tidak akan pernah jalan
// d. Semua jawaban salah
```

// 11. Berikut ini yang merupakan falsy Javascript yaitu?

```
// a. " "
// b. 2020
// c. []
// d. -0
```

// 12. Perhatikan code berikut! Hasil dari kedua console.log adalah?

```
/*
```

```
let platNomor = "B 1234 A"
console.log(String(platNomor))
console.log(Number(platNomor))
*/
// a. "B 1234 A" & 1234
// b. "B 1234 A" & NaN
// c. "BA" & 1234
// d. Undefined & NaN
```

```
// 13. Berikut pernyataan yang tepat mengenai looping while dan do while yaitu?
// a. While menjalankan action terlebih dahulu, dilanjutkan dengan memeriksa kondisi
// b. While akan menjalankan action apabila kondisinya tidak terpenuhi
// c. Do While menjalankan action terlebih dahulu, dilanjutkan dengan memeriksa kondisi
// d. Do While tidak membutuhkan sebuah kondisi untuk menjalankan action nya
```

```
// 14. Hasil dari code berikut apabila dijalankan yaitu?
/*
const nilai = 0
const jawaban = 'acdab'
const kunciJawaban = 'abdab'
for(let i=0; i<jawaban.length; i++){
  let result = false // Define di dalam local scope

  if(jawaban[i] === kunciJawaban[i]){
    result = true
  }

  if(result){
    nilai += 20
  }
}
console.log(result) // Kita mencoba mengakses di outside scope nya
*/
// a. ReferenceError: result is not defined
// b. ReferenceError: Assignment to constant variables
// c. SyntaxError: Identifier 'result' has already been declared
// d. Undefined
```

```
// 15. Manakah code dibawah ini yang menghasilkan data bertipe Boolean yang bernilai true?  
// a. console.log(1 == '1')  
// b. console.log(null === undefined)  
// c. console.log(10 === '10')  
// d. console.log(String(20) === "20")
```

```
// 16. Hasil dari console.log di bawah ini yaitu?
```

```
/*  
    let printLn = "Hello"  
  
    if(printLn){  
        printLn += ", World!"  
    }else{  
        printLn = "Hai, Dunia!"  
    }  
*/  
// a. "Hello, World!"  
// b. ", World!"  
// c. "Hai, Dunia"  
// d. Semua jawaban salah
```

```
// 17. Berikut ini yang merupakan comparison operator, kecuali?
```

```
// a. =>  
// b. <=  
// c. ==  
// d. ===
```

```
// 18. Berikut ini yang merupakan tipe data yang bersifat immutable, kecuali?
```

```
// a. String  
// b. Number  
// c. Array  
// d. NaN
```

```
// 19. Perhatikan code berikut! Apa yang akan terjadi apabila program tersebut dijalankan?
```

```
/*  
    let campus = "Purwadhika BSD"
```

```

{
  var campus = "Purwadhika JKT"
}

console.log(campus)
*/
// a. Purwadhika BSD
// b. Purwadhika JKT
// c. Error (Identifier 'campus' has already been declared)
// d. Error (Assignment to constant variable)

```

// 20. Perhatikan code berikut! Apa yang akan terjadi apabila program dijalankan?

```

/*
  let nilaiModul = 70

  if(nilaiModul > 50){
    console.log("Anda Remedial")
  }if(nilaiModul > 60){
    console.log("Anda Hampir Tidak Remedial")
  }if(nilaiModul >= 70){
    console.log("Anda Lulus")
  }
*/
// a. Anda Remedial
// b. Anda Hampir Tidak Remedial
// c. Anda Lulus
// d. Semua jawaban salah

```

```

let nilaiModul = 70

if(nilaiModul > 50){
  console.log("Anda Remedial")
}

if(nilaiModul > 60){
  console.log("Anda Hampir Tidak Remedial")
}

if(nilaiModul >= 70){
  console.log("Anda Lulus")
}

```

```

/*
  let const = 123
  const < 123? const = 321 : console.log("Hello")
*/

// 1. Perhatikan code berikut!
/*
  let liga1 = ["Persib", "Persija", "Bali United"]
  liga1[50] = "PSM"
  console.log(liga1.length)
  console.log(liga1.indexOf("PSM"))
*/
// Hasil yang tepat dari kedua console.log diatas adalah?
//   a. 51 & 51
//   b. 51 & 50 x
//   c. 49 & 50
//   d. 50 & 49

```

```

// 2. Syntax manakah yang merupakan ternary operator?
//   a. 1 => 2? return true : return false
//   b. 10 = 10? console.log("true") : console.log("false")
//   c. "ABC" !== "abc"? return 1 :: return 0
//   d. 100 < 101? true : false x

```

```

// 3. Perhatikan program berikut!
/*
  hasilKali(5)
  function hasilKali(num){

```

```

    console.log(10 * 100)
  }

  hasilBagi()
  let hasilBagi = function(num){
    return num % num
  }
*/
// Apa yang akan terjadi apabila program diatas dijalankan?
// a. Function hasilKali error, karena tidak menggunakan parameter num
// b. return num undefined, karena tidak menerima argument
// c. Function hasilBagi error, karena di call sebelum di declare x
// d. console.log hasilKali undefined, karena tidak menggunakan parameter num

```

```

// 4. Perhatikan code berikut!
/*
  let point = '100'
  point+1 // Tidak merubah apapun
  point++
  point++
  point--
  point = point + '1'
*/
// Hasil console.log(point) yaitu?
// a. 1021
// b. 1011 x
// c. 103
// d. 102

```

```

// 5. Hasil dari kedua console.log pada program berikut yaitu?
/*
  let dataSiswa = {
    nama: "Aisyah",
    umur: 20
  }

  console.log(dataSiswa.nama)
  console.log(dataSiswa.hobi)
*/
// a. Aisyah & ReferenceError: property hobi is not defined

```



```
// b. Aisyah & Undefined x
// c. Budi & Null
// d. Budi & ""
```

// 6. Hasil dari console di bawah ini yaitu?

```
/*
  let cars = ["Honda", "Toyota", "Wuling"]
  console.log(...cars)
*/
// a. Honda Toyota Wuling x
// b. ["Honda", "Toyota", "Wuling"]
// c. Honda, Toyota, Wuling
// d. "Honda, Toyota, Wuling"
```

// 7. Perhatikan code di bawah ini!

```
/*
  let dataToSend = ["Haekal", 20, "FE Dev"]

  function printLn(a, b, c){
    console.log(`
      Saya ${a},
      Umur ${b},
      Saya sebagai ${c}.
    `)
  }

  printLn(...dataToSend)
*/
// Hasil program diatas akan menghasilkan?
// a. Saya Haekal, Umur 20, Saya sebagai FE Dev. x
// b. TypeError: Found non-callable @@iterator
// c. TypeError: Assignment to constant variable.
// d. ReferenceError: dataToSend not iterable
```

// 8. Pernyataan yang tepat mengenai synchronous yaitu?

```
// a. Untuk menjalankan baris code secara parallel
// b. Untuk menjalankan tugas yang mungkin memerlukan waktu tanpa harus memblokir tugas berikutnya
```

```
// c. Untuk menjalankan 1 tugas atau lebih secara bersamaan dalam kurun waktu yang bersamaan
// d. Untuk menjalankan operasi berikutnya setelah operasi sebelumnya selesai dijalankan x
```

```
// 9. Perhatikan data array berikut ini:
```

```
/*
  let data = [
    true, undefined, [1, 2, [false, ['Lynette', 'Wahyu'], 3]],
    "Patrick", NaN, 100, [3, "Joko", [{school: "Purwadhika"}]],
    0, -100, `Lucas`, ['Javascript']
  ]
*/
// Untuk mendapatkan "Purwadhika", maka console.log yang tepat yaitu?
// a. console.log(data[6][3][1].school)
// b. console.log(data[6][2][0].school) x
// c. console.log(data[6][2][1].object(school))
// d. Error, karena array tidak dapat menampung tipe data NaN & undefined
```

```
// 10. Perhatikan syntax berikut!
```

```
/*
  const arr1 = ["-", "---", "-"]
  arr1.push("----")
*/
// Apakah yang akan terjadi apabila program dijalankan?
// a. TypeError: Assignment to constant variable
// b. ReferenceError: Cannot access 'arr1' before initialization
// c. TypeError: Declare before initialization
// d. Semua Jawaban Salah x
```

```
// 11. Perhatikan looping berikut!
```

```
/*
  let str = 'Garuda di dadaku!'.split(" ")

  for(let i=0; i < str.length; i += 1){
    console.log(str[i])
    str.push(str[i])
  }
*/
```

// Code tersebut ketika dijalankan ternyata error, apa yang menyebabkan error tersebut?
// a. Salah syntax pada for loop
// b. Kondisi step pada for loop tidak bisa i += 1
// c. String pada variabel str tidak bisa diberi method .split(' ') secara langsung
// d. Terjadi infinite loop karena str.push(str[i]) x

// 12. Berikut yang termasuk truthy Javascript yaitu?
// a. [] x
// b. ""
// c. ``
// d. NaN

// 13. Berikut ini yang merupakan build in function pada Javascript kecuali?
// a. toLocaleString()
// b. constructor()
// c. toFixed()
// d. new Date() x

// 14. Hasil dari console.log pada program berikut adalah?
/*
 let number = 10
 let number = 100
 console.log(number)
*/
// a. 100
// b. 10
// c. TypeError: Assignment to constant variable
// d. SyntaxError: Identifier number has already been declared x

// 15. Jika ingin menjalankan program javascript pada file html, maka tag element yang digunakan adalah?
// a. <html></html>
// b. <jscript></jscript>
// c. <script></script> x
// d. <code></code>

// 16. Dibawah ini manakah penamaan variable yang paling tepat?

- // a. function
- // b. 10people_
- // c. people-name
- // d. People!

// 17. Dibawah ini manakah variable yang menghasilkan value undefined apabila variable tersebut di akses terlebih dahulu sebelum didefine?

- // a. var x
- // b. let
- // c. const
- // d. Semua jawaban benar

// 18. Manakah operator yang menghasilkan Boolean bernilai true, apabila komparasi kedua data memiliki value yang berbeda?

- // a. !== x
- // b. ===
- // c. ==
- // d. &&

// 19. Program di bawah ini akan dijalankan sebanyak berapa kali?

/*

let point = 10

```
function A(){  
  point += 5  
  return(point)  
  c()  
}
```

```
let b = (newPoint) => {  
  point += newPoint  
}
```

```
let c = function(){  
  console.log("Hello, World!")  
}
```

```

    A()
  }

  console.log(A())
*/
// a. 1x x
// b. 2x
// c. Infinite
// d. Tidak running

```

// 20. Hasil dari console.log pada looping berikut adalah?

```

/*
  for(let i=10; i>=0; i--){
    if(i%2 === 0 || i !== 0){
      console.log(i)
    }
  }
*/
// a. 10, 8, 6, 4, 2, 0 x
// b. 10, 8, 6, 4, 2
// c. 9, 7, 5, 3, 2, 1
// d. 7, 5, 3, 2, 1

```

// 1. Sebutkan macam-macam comparison operator dan logical operator!

// Comparison operator: >, >=, <, <=, ==, ===, !==

// Logical operator: && AND, || OR, ! NOT

// 2. Apakah perbedaan antara AND, OR dan juga NOT?

// AND: Kondisi keduanya harus true ---> True

// OR: Salah 1 kondisi bernilai true ---> True

// NOT: Negasi

// 3. Perhatikan code berikut! Apa yang akan terjadi apabila program berikut dijalankan?

```

/*
  function GetArr(arr){ ---> Undefined
    return arr.forEach(value => {
      return value
    })
  }

  function GetValue(arr){ ---> ["A", null, true, undefined, 20]
    return arr.map(value => {
      return value
    })
  }

```

```

}

let arr = ["A", null, true, undefined, 20]
console.log(GetArr(arr))
console.log(GetValue(arr))
*/
// 4. Sebutkan perbedaan variable yang di declare dengan menggunakan syntax var, let dan juga const!
// var
// - Tidak mengenal scope
// - Dapat di duplicate
// - Value nya dapat diubah
// let
// - Mengenal scope
// - Tidak dapat di duplicate
// - Value nya dapat diubah
// const
// Mirip dengan let, hanya value nya tidak bisa diubah
// 5. Sebutkan perbedaan antara looping while, do while dan juga for loop!
// While: Memeriksa kondisi, baru jalan
// Do While: Jalan sekali, baru memeriksa kondisi
// For Loop: Sama seperti while
// 6. Sebutkan 5 build in method yang ada di Javascript!
// .toString(), .parseInt(), .parseFloat(), .toLowerCase(), .toUpperCase()
// 7. Perhatikan code berikut! Apakah hasil console.log program di bawah ini?
/*
var name = "Defryan"
let newName = name
newName = "Ryan"
console.log(name) // "Defryan"
console.log(newName) // "Ryan"

var listSiswa = ["Stephen", "Nathan", "Syahrul"]
let newListSiswa = [...listSiswa] // ["Stephen", "Nathan", "Syahrul"]
newListSiswa[0] = "Fauzan" // ["Fauzan", "Nathan", "Syahrul"]
console.log(listSiswa) // ["Stephen", "Nathan", "Syahrul"]
console.log(newListSiswa) // ["Fauzan", "Nathan", "Syahrul"]
*/
// 8. Sebutkan perbedaan antara mutable dan immutable!
// - Mutable: Data aslinya bisa berubah
// - Immutable: Data aslinya tidak bisa diubah
// 9. Sebutkan 3 contoh truthy dan falsy Javascript!
// Truthy: [], 10, {}
// Falsy: "", null, undefined, 0, dll

```

```

// 10. Perhatikan code berikut ini! Tuliskan dalam bentuk ternary operator!
/*
    if(10 !== "10"){
        return 10
    }else if(10 !== 10){
        return "10"
    }else{
        return 10
    }
*/
// 10 !== "10"? return 10 : 10 !== 10? return "10" : return 10;
// 11. Sebutkan perbedaan break dan juga continue!
// Break: Untuk menghentikan proses pengulangan / program
// Continue: Untuk men-skip program di bawahnya
// 12. Sebutkan perbedaan method push, unshift, shift dan juga pop!
// Push: Masukan data dalam array di paling akhir
// Unshift: Masukan data di paling awal
// Shift: Menghapus data di paling awal
// Pop: Menghapus data di paling akhir
// 13. Perhatikan code berikut! Manakah penadeklarasian variable yang tidak sesuai
// dengan aturan JS?
/*
    let $studentLists = ["Rafli", "Dhito", "Vero"] TRUE
    var KampusPWD = "Bsd, Jkt, Btm" TRUE
    const _e_commerce = null TRUE
    let programmingLanguage = { Javascript, PHP, Java } FALSE
    var comp4ny = "Bithealth, Tokopedia, Werkdone" TRUE
*/
// 14. Sebutkan macam-macam data structure dan juga cara kerjanya!
// - Stack: LIFO
// - Queue: FIFO
// - Set: Data nya tidak boleh duplicate
// - Linkedlist: Menunjuk data sebelum/sesudahnya
// - Hashmap: Mirip object
// 15. Apakah perbedaan antara forEach, .map dan juga .filter?
// 16. Jelaskan secara singkat proses synchronous dan asynchronous!
// - Synchronous: Program dijalankan secara berurutan
// - Asynchronous: Program dijalankan secara paralel
// 17. Apa itu hoisting?
// Program kita ditarik ke atas
// 18. Buatlah contoh program dengan menggunakan indirect callback function!
//
// 19. Perhatikan data berikut! Tuliskan syntax yang tepat untuk menambahkan
// buah "Pear" dan menghapus buah "Apel" menggunakan method splice!

```

```
// let arrBuah = ["Pisang", "Apel", "Jeruk", "Mangga", "Apel"]
// arrBuah.splice(1, 1, "Pear")
// 20. Sebutkan macam-macam tipe data primitive dan non primitive!
// Primitive: String, Number, Null, Undefined, Boolean
// Non-primitive: Object & Array
```

```
let arr = [1, 2, 3]
let newData = arr.forEach(value => { value }) // forEach hanya me-looping namun tidak
menghasilkan array baru
let newDataSatu = arr.map(value => { return value }) // map me-looping sekaligus menghasilkan
array baru
console.log(newData)
console.log(newDataSatu)
```

```
// 1. Perhatikan code berikut!
```

```
/*
  const dataSiswa = {
    name: "Amel",
    umur: 22,
    noHp: [08123598733, "08523345976"]
  }
  console.log(dataSiswa.name)
  console.log(dataSiswa.nohp)
*/
```

```
// Hasil dari perintah console.log diatas adalah?
// a. Amel and 'ReferenceError: key nohp is not defined'
// b. Amel and null
// x Amel and undefined ---> Karena case sensitive (nohp !== noHp)
// d. Amel and 'TypeError: nohp is not props'
```



```
// 2. Apakah perbedaan antara syntax let dan const?  
// a. Value dari variable let tidak dapat diubah, sedangkan const dapat diubah  
// b. Variable let memiliki aturan scope, sedangkan const tidak memiliki aturan scope  
// c. Variable const dapat diduplicate, sedangkan let tidak dapat di duplicate  
// x Value dari variable const tidak dapat diubah, sedangkan let dapat diubah x
```

```
// 3. Hasil console.log(typeof null) dan juga console.log(type of 2302) adalah?  
// a. string & number  
// b. number & number  
// x object & number  
// d. null & number
```

```
// 4. Perhatikan code berikut!
```

```
/*  
    let std = "GBK"  
    std += " JKT"  
    std = "GBT SBY"  
    console.log(std)
```

```
*/  
// a. "GBK"  
// b. " JKT"  
// x "GBT SBY"  
// d. "GBK JKT"
```

```
// 5. Perbedaan dari .forEach dengan .map adalah?
```

```
// x forEach tidak menghasilkan array baru, sedangkan map menghasilkan array baru
// b. Looping forEach dapat dihentikan dengan syntax break, sedangkan looping map
//   dapat dihentikan dengan syntax return
// c. Looping map digunakan untuk me-looping saja, sedangkan looping forEach digunakan
//   untuk mendapatkan output berupa array baru
// d. Semua jawaban salah
```

```
// 6. Pendeklarasian yang sesuai dengan aturan JS adalah?
// a. const password = @abc123#
// x let school_Name = "Purwadhika"
// c. var Object = { name: "Andien" }
// d. let nama-depan = "Budi"
```

```
// 7. Data structure yang menggunakan konsep FIFO adalah?
// a. Stack
// x Queue
// c. Linkedlist
// d. Hashmap
```

```
// 8. Bagaimana cara untuk mengakses property output pada data berikut ini?
/*
    const data = [
        null, () => [{ output: 'Hai, aku disini!' }], 100, undefined, true
    ]
*/
// x data[1]()[0]['output'] // Bracket || pakai dot ---> data[1]()[0].output
// b. data[1][0].output
// c. data[1]().output
// d. data[1]()=>[0].output
```

// 9. Perhatikan program berikut ini!

```
/*  
    let discount = 75  
    discount *= 1 // 75  
    discount++ // 76  
    discount += '1' // 761  
*/  
// Hasil console.log(discount) adalah?  
// a. 7521  
// x 761  
// c. 77  
// d. 751
```

// 10. Perhatikan looping berikut!

```
/*  
    do{  
        let poin = 77 // Harus pakai var  
        poin+=1  
        console.log(poin)  
        poin--  
    }while(poin <= 80)  
*/  
// Apakah kesalahan pada program di atas?  
// a. Kondisi step pada exit way tidak bisa menggunakan += 1  
// b. Terjadi infinite loop karena poin--  
// c. Kesalahan pada penamaan variable  
// x Variable poin tidak dapat digunakan di dalam while condition
```

// 11. Perhatikan syntax berikut!

```
/*
```

```

    let noAntrian = [11, 13, 15, 12, 14]
    noAntrian.push(12) // [11, 13, 15, 12, 14, 12]
    noAntrian.sort() // [11, 12, 12, 13, 14, 15]
    noAntrian.pop() // Menghapus data terakhir ---> [11, 12, 12, 13, 14]
*/
// Hasil akhir dari noAntrian adalah?
// x [11, 12, 12, 13, 14]
// b. [12, 12, 13, 14, 15]
// c. [11, 13, 15, 12, 14]
// d. [11, 12, 13, 14, 15]

```

```

// 12. Perhatikan syntax berikut ini!
/*
    function cetakTiket(tiket, callback){
        return callback(tiket)
    }

    cetakTiket('Kereta', Print)
*/
// Apakah yang akan terjadi apabila program diatas dijalankan?
// x TypeError: Callback is not a function
// b. ReferenceError: Callback is not defined
// c. 'Kereta'
// d. Semua jawaban salah

```

```

// 13. Berikut ini pernyataan yang tidak tepat mengenai tipe data array adalah?
// a. Memiliki property .length saja
// b. Menampung semua tipe data termasuk array itu sendiri
// c. Masing-masing data memiliki index penomoran
// x Tidak dapat menampung object dan juga function

```

```

// 14. Hasil dari console.log(String(2302) === "2302") adalah?
// x True
// b. False
// c. ReferenceError
// d. TypeError

```

```
// 15. Perhatikan code berikut!  
/*  
    let printKuy = "Selamat"  
    if("0" !== "no!"){ // true  
        printKuy += " Apa Hayo?"  
    }  
*/  
// Hasil dari console.log(printKuy) di akhir program adalah?  
// a. " Apa Hayo?"  
// b. "Selamat"  
// x "Selamat Apa Hayo?"  
// d. "Selamat Apa Hayo"
```

```
// 16. Perhatikan function berikut!  
/*  
    let UppercaseStr = (txt) => {  
        let text = txt  
        return text  
    }  
    UppercaseStr("Ya")  
    console.log(text)  
*/  
// Apa hasil dari console.log program diatas?  
// a. Undefined  
// b. Ya  
// x ReferenceError: text is not defined  
// d. Null
```

```
// 17. Perhatikan code berikut!  
/*
```

```

let kelas = '2302JCWD'
kelas = Number(kelas)
kelas = String(kelas)
*/
// Apakah hasil dari console.log(kelas) di baris akhir dan apa type of dari kelas?
// a. Null & String
// x NaN & String
// c. 2302 & String
// d. 2302 & Number

```

```

// 18. let clothing = ["Erigo", "Wadezig", "UNKL347"]
//   clothing.splice(2, 0, "Reclays", "Screamous") // Erigo, Wadezig, Reclays, Screamous, UNKL
//   clothing.shift() // Wadezig, Reclays, Screamous, UNKL
//   console.log(clothing)
// a. ["Wadezig", "UNKL347", "Reclays", "Screamous"]
// b. ["Erigo", "Wadezig", "UNKL347", "Reclays"]
// x ["Wadezig", "Reclays", "Screamous", "UNKL347"]
// d. ["Erigo", "Wadezig", "UNKL347", "Reclays"]

```

```

// 19. Perhatikan program berikut!
/*
    let klub = "Persib"
    let std = "GBLA"

    if(!klub && !std){
        console.log("Data Tidak Lengkap")
    }else if(klub && !std){
        console.log("Data Kurang Lengkap")
    }else if(!klub && std){
        console.log("Data Masih Kurang Lengkap")
    }else{
        console.log("Data Lengkap")
    }
*/
// Hasil dari program diatas adalah?

```

```
// a. Data Tidak Lengkap
// b. Data Kurang Lengkap
// c. Data Masih Kurang Lengkap
// x Data Lengkap
```

// 20. Berikut ini syntax yang dapat dilakukan hoisting yaitu?

```
// a. var & let myFunc = function(){ ... }
// b. let & function NewFunction(){ ... }
// c. const & const createFunction = () => { ... }
// d. var & var ourFunction = () => { ... }
```

// 1.

```
/*
  let Halo = "Aku Tito"
  let halo = Halo.toUpperCase()
  console.log(Halo);
*/
```

// Apa hasil dari console.log tersebut? Karena javascript adalah sensitive case

```
// a. Perang Sticker
// b. AKU TITO
// x Aku Tito
// d. Error
```

// 2.

```
/*
  let a = "0"
  let b = Number(a)
  let hasil = a+b // "0" + 0
  console.log(hasil); "00"
*/
```

// Apa hasil dari console.log tersebut?

```
// a. undefined
// x 00
// c. NaN
// d. 0
```

```
// 3.
/*
  let a = 3
  let b = 3
  if(a == b){
    console.log(a**3);
  }else if(a === b){
    console.log(a**2);
  }else{
    console.log("Ngaco");
  }
*/

// Console.log manakah yang akan dijalankan?
// x 27
// b. 9
// c. "Ngaco"
// d. 3
```

```
// 4.
/*
  function Babayo(num){
    if(2+1**1 === num){ // (3 === 3)
      return "Oyapohh"
    }else if(num == "3"){
      return "Aooo"
    }
  }
  console.log(Babayo(3));
*/

// Hasil apa yang didapat dari function tersebut?
// a. Undefined
// x "Oyapohh"
// c. "Aooo"
// d. "Izin pipis"
```

```
// 5.
/*
  let a = [1, 3, 5, 6]
  let hasil = 0 // 1
```



```

for(let i=0 ; i<a.length ; i++){
  if(hasil == 0){
    hasil+=a[i]
  }else if(hasil != 0){
    hasil*=a[i]
  }
}
console.log(hasil);
*/

```

// Agar value hasil tidak selalu bernilai 0 dan dapat menghasilkan hasil 90, maka kita harus mengisi baris kosong dengan?

```

// a. hasil == a[i]
// b. hasil *= a[i]
// x hasil += a[i]
// d. hasil /= a[i]

```

// 6.

```

/*
let huruf = ["A", "B", "C", "D", "E"]
huruf = ...
console.log(huruf);
*/

```

// Untuk mendapatkan hasil variable huruf ['A', 'B', 'C'], maka kita akan memakai method apa?

```

// a. huruf.splice(0, 3)
// b. huruf.concat(0, 3)
// x huruf.slice(0, 3)
// d. huruf.delete(0, 3)

```

// 7.

```

/*
let a = [
  [{nama: "Masdef", Umur: 5}],
  {nama: "Tito", Umur: 12},
  {nama: "Ka Youke", Umur: 10}
]
*/

```

// console.log yang tepat untuk mendapatkan Umur 5 adalah

```

// a. console.log(a[0][0].umur);
// b. console.log(a[0][].umur);

```

```
// c. console.log(a[0][0][0].Umur);  
// x console.log(a[0][0].Umur);
```

```
// 8.  
// Sifat pada Queue adalah? Queue Ngantri  
// a. VIVO  
// b. TITO  
// c. LIFO  
// x FIFO
```

```
// 9.  
// Method yang dapat membuat array baru adalah  
// a. .forEach()  
// b. .pilter()  
// x .map()  
// d. .mep()
```

```
// 10.  
// Yang merupakan bagian dari promise yang benar adalah kecwali?  
// x .finally() Tetap terhubung tetapi menghasilkan sukses atau error saja  
// b. .then() Tetap terhubung tetapi menghasilkan sukses atau error saja  
// c. .catch() Tidak terhubung samsek atau kendala server  
// d. .finally() Selalu berjalan
```

```
// 1000000000000000
```