// course search for navbar

Tila	Palautettu
Aloitettiin	maanantai 6. lokakuuta 2025, 09.47
Valmis	maanantai 6. lokakuuta 2025, 10.28
Suorituskerran kesto	41 min 25 sekuntia
Arviointi	**0,00** pistettä maksimista 14,00 (**0**%)

Kysymys 1

Väärin

Pisteet 0,00 kokonaispisteistä 2,00

Create a JavaScript function called `calculateCylinderVolume` that calculates the volume of a right circular cylinder. The function should take two parameters: the radius of the base ('r') and the height of the cylinder ('h'). The function should return the calculated volume. You can use the formula for the volume of a cylinder, which is $V = \pi r^2 h$

You can get PI from Math class.

Provide the code for the `calculateCylinderVolume` function.

Esimerkiksi:

Testi	Tulos
const volume = calculateCylinderVolume(10, 90); console.log(volume);	28274.33388230814

Vastaus: (rangaistus: 0 %)

```
function calculateCylinderVolume() {

    const volume = calculateCylinderVolume(pi*r^2*h);
    console.log(volume$());
}
```

	Testi	Odotettu	Saatu tulos	

✗	const volume = calculateCylinderVolume(10, 90);
	console.log(volume);
		28274.33388230814	***Run error***
			__tester__.js:3
			  const volume = calculateCylinderVolume(pi*r^2*h);
			  ^
			
			ReferenceError: pi is not defined
			  at calculateCylinderVolume (__tester__.js:3:44)
			  at Object.<anonymous> (__tester__.js:9:16)
			  at Module._compile (node:internal/modules/cjs/loader:1356:14)
			  at Module._extensions..js (node:internal/modules/cjs/loader:1414:10)
			  at Module.load (node:internal/modules/cjs/loader:1197:32)
			  at Module._load (node:internal/modules/cjs/loader:1013:12)
			  at Function.executeUserEntryPoint [as runMain]
			(node:internal/modules/run_main:128:12)
			  at node:internal/main/run_main_module:28:49
			
			Node.js v18.19.1
				✗

Testaaminen lopetettiin virheen vuoksi.
Ohjelmasi täytyy läpäistä kaikki testit saadaksesi pisteitä.

Näytä erot

Malliratkaisu (Nodejs)

```
function calculateCylinderVolume(r, h) {
  return Math.PI * r * r * h;
}
```

Väärin

Pisteet tälle palautukselle: 0,00/2,00.

Kysymys 2

Väärin

Pisteet 0,00 kokonaispisteistä 2,00

Write a JavaScript function called `calculateDiscountRate` that takes a numerical `purchaseAmount` as an argument and returns the corresponding discount rate based on the following tiered structure:

- If the purchase amount is 500 **or more**, return 0.2
- If the purchase amount is between 250 **and** 499 (inclusive), return 0.1
- If the purchase amount is between 100 **and** 249 (inclusive), return 0.05
- If the purchase amount is **below** 100, return 0

Provide the code for the `calculateDiscountRate` function using a standard `if-else` statement structure.

Write **only** the function, nothing else.

Esimerkiksi:

Testi	Tulos
console.log(calculateDiscountRate(550));	0.2
console.log(calculateDiscountRate(499));	0.1
console.log(calculateDiscountRate(250));	0.1

Vastaus: (rangaistus: 0 %)

```
function calculateDiscountRate () {
    console.log(calculateDiscountRate(550));
     === w => ? w =< 500;
        return 0.2
    if-else purchaseAmount =< 250;
        return 0.1
    if-else purchaseAmount =< 100;
        return 0.05
    else
        return 0
}
```

	Testi	Odotettu	Saatu tulos	
⊗	console.log(calculateDiscountRate(550));	0.2	***Run error*** __tester__.js:3 === w => ? w =< 500; ^^^ SyntaxError: Unexpected token '===' at internalCompileFunction (node:internal/vm:73:18) at wrapSafe (node:internal/modules/cjs/loader:1274:20) at Module._compile (node:internal/modules/cjs/loader:1320:27) at Module._extensions..js (node:internal/modules/cjs/loader:1414:10) at Module.load (node:internal/modules/cjs/loader:1197:32) at Module._load (node:internal/modules/cjs/loader:1013:12) at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:128:12) at node:internal/main/run_main_module:28:49 Node.js v18.19.1	⊗

Testaaminen lopetettiin virheen vuoksi.
Ohjelmasi täytyy läpäistä kaikki testit saadaksesi pisteitä.

Näytä erot

Malliratkaisu (Nodejs)

```
function calculateDiscountRate(purchaseAmount) {
    if (purchaseAmount >= 500) {
        return 0.2;
    } else if (purchaseAmount >= 250) {
        return 0.1;
    } else if (purchaseAmount >= 100) {
        return 0.05;
    } else {
        return 0;
    }
}
```

Väärin

Pisteet tälle palautukselle: 0,00/2,00.

Kysymys 3

Väärin

Pisteet 0,00 kokonaispisteistä 1,00

Write a JavaScript function *calculateCube* that calculates the volume of a cube, taking the length of one edge as its argument.

Volume (V) of a cube: $V = s^3$

Esimerkiksi:

Testi	Tulos
console.log(calculateCube(2))	8

Vastaus: (rangaistus: 0 %)

Tyhjennä vastaus

```
function calculateCube() {
    calculateCube{
    console.log(('w'))

}
```

	Testi	Odotettu	Saatu tulos	
⊗	console.log(calculateCube(2))	8	***Run error*** __tester__.js:2 calculateCube{ ^ SyntaxError: Unexpected token '{' at internalCompileFunction (node:internal/vm:73:18) at wrapSafe (node:internal/modules/cjs/loader:1274:20) at Module._compile (node:internal/modules/cjs/loader:1320:27) at Module._extensions..js (node:internal/modules/cjs/loader:1414:10) at Module.load (node:internal/modules/cjs/loader:1197:32) at Module._load (node:internal/modules/cjs/loader:1013:12) at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:128:12) at node:internal/main/run_main_module:28:49 Node.js v18.19.1	⊗

Testaaminen lopetettiin virheen vuoksi.
Ohjelmasi täytyy läpäistä kaikki testit saadaksesi pisteitä.

Näytä erot

▾ **Malliratkaisu (Nodejs)**

```
function calculateCube(edge){
    return edge ** 3;
```

Väärin

Pisteet tälle palautukselle: 0,00/1,00.

Kysymys 4

Ei vastattu

Pisteet 0,00 kokonaispisteistä 4,00

Create a JavaScript function, `filterInventoryByThreshold`, that processes an array of product objects based on a variable stock level requirement.

Input Details:

1. `inventory`: An array of product objects. Each object has the properties `name` (string) and `stock` (number).
    - *Example:* `{ name: "Laptop", stock: 15 }, { name: "Mouse", stock: 50 }, { name: "Keyboard", stock: 8 }`
2. `minStockThreshold`: An integer defining the minimum stock level required for a product to be included.

Function Requirement:

The function must return a new array containing only the `name` of the products whose `stock` is strictly greater than or equal to the provided `minStockThreshold`.

Flexibility Constraint:

The function must be designed to be robust and flexible. Specifically, provide the code that correctly calculates the filtered list, ensuring that it accurately reflects the results even if the `minStockThreshold` value is **changed on subsequent calls** (e.g., calling the function first with a threshold of 10, and then immediately with a threshold of 30 should produce correct, inde results).

Provide **only** the code for this function.

Esimerkiksi:

Testi	Tulos

const warehouseInventory = [
{ name: "Laptop Pro", stock: 15 },
{ name: "4K Monitor", stock: 35 },
{ name: "Wireless Mouse", stock: 50 },
{ name: "Ergo Keyboard", stock: 8 },
{ name: "HD Webcam", stock: 20 },
{ name: "USB Hub", stock: 10 }
];
 	
const result = filterInventoryByThreshold(warehouseInventory, 10);
console.log(result);
	[
	'Laptop Pro',
	'4K Monitor',
	'Wireless Mouse',
	'HD Webcam',
	'USB Hub'
	]

Vastaus: (rangaistus: 0 %)

```
1
```

	Testi	Odotettu	Saatu tulos

⊗	const warehouseInventory = [
	{ name: "Laptop Pro", stock: 15 },
	{ name: "4K Monitor", stock: 35 },
	{ name: "Wireless Mouse", stock: 50 },
	{ name: "Ergo Keyboard", stock: 8 },
	{ name: "HD Webcam", stock: 20 },
	{ name: "USB Hub", stock: 10 }
	];
	
	const result = filterInventoryByThreshold(warehouseInventory, 10);
	console.log(result);
		[
		'Laptop Pro',
		'4K Monitor',
		'Wireless Mouse',
		'HD Webcam',
		'USB Hub'
		]
			Run error
			__tester__.js:2
			const warehouseInventory = [
			^^^^^
			
			SyntaxError: Unexpected token 'const'
			at internalCompileFunction (node:internal/vm:73:18)
			at wrapSafe (node:internal/modules/cjs/loader:1274:20)
			at Module._compile (node:internal/modules/cjs/loader:1320:27)
			at Module._extensions..js (node:internal/modules/cjs/loader:1414:10)
			at Module.load (node:internal/modules/cjs/loader:1197:32)
			at Module._load (node:internal/modules/cjs/loader:1013:12)
			at Function.executeUserEntryPoint [as runMain]
			(node:internal/modules/run_main:128:12)
			at node:internal/main/run_main_module:28:49
			
			Node.js v18.19.1

Testaaminen lopetettiin virheen vuoksi.
Ohjelmasi täytyy läpäistä kaikki testit saadaksesi pisteitä.

Näytä erot

Malliratkaisu (Nodejs)

```
function filterInventoryByThreshold(inventory, minStockThreshold) {
    const filteredProducts = inventory.filter(product => {
        return product.stock >= minStockThreshold;
    });

    const productNames = filteredProducts.map(product => product.name);

    return productNames;
}
```

Väärin

Pisteet tälle palautukselle: 0,00/4,00.

Kysymys 5

Ei vastattu

Pisteet 0,00 kokonaispisteistä 3,00

Write a JavaScript function called `transformToFullName` that takes an array of objects as its input. Each object represents a person with `firstName` and `lastName` properties. The function must return a **new array** of strings, where each string is the person's full name (e.g., "John Doe"). The original array of objects should **not** be modified.

Provide **only** the code for the `transformToFullName` function.

Esimerkiksi:

Testi	Tulos

const employees = [
{ firstName: "Ada", lastName: "Lovelace" },
{ firstName: "Grace", lastName: "Hopper" },
{ firstName: "Alan", lastName: "Turing" }
];
const fullNames = transformToFullName(employees);
console.log(employees);
console.log(fullNames);
	[
	{ firstName: 'Ada', lastName: 'Lovelace' },
	{ firstName: 'Grace', lastName: 'Hopper' },
	{ firstName: 'Alan', lastName: 'Turing' }
	]
	['Ada Lovelace', 'Grace Hopper', 'Alan Turing']

Vastaus: (rangaistus: 0 %)

```
1
```

▾ **Malliratkaisu (Nodejs)**

```
function transformToFullName(people) {
  return people.map(person => `${person.firstName} ${person.lastName}`);
}
```

Kysymys 6

Ei vastattu

Pisteet 0,00 kokonaispisteistä 2,00

Write a JavaScript function '**filterOddNumbers**' that filters an array to include only odd numbers and returns a new array with odd numbers.

Esimerkiksi:

Testi	Tulos

const filteredArray = filterOddNumbers([1, 2, 3, 4, 5, 6]);
 	
// Convert the filtered array to a string for comparison
const resultString = filteredArray.join(',');
console.log(resultString);
	1,3,5

Vastaus: (rangaistus: 0 %)

```
1
```

▾ **Malliratkaisu (Nodejs)**

```
function filterOddNumbers(numbers) {
    return numbers.filter(number => number % 2 !== 0);
}
```