

# DWA\_03.4 Knowledge Check\_DWA3.1

---

1. Please show how you applied a Markdown File to a piece of your code.

## # Shipping Calculator

This script calculates the total cost of items and determines the shipping cost based on the location. It also checks if free shipping applies to the order.

### ## Constants

- `RSA\_SHIPPING`: Shipping cost for orders in South Africa (RSA) - \$400.
- `NAM\_SHIPPING`: Shipping cost for orders in Namibia (NAM) - \$600.
- `OTHER\_SHIPPING`: Shipping cost for orders in other locations - \$800.
- `FREE\_WARNING`: Warning message for free shipping eligibility.
- `BANNED\_WARNING`: Warning message for countries where shipping is not available.

### ## Variables

- `shoes`: Cost of shoes - \$300.
- `toys`: Cost of toys (100 each) - \$500.
- `shirts`: Cost of shirts (150 each) - \$0 (assumed no shirts purchased).
- `batteries`: Cost of batteries (35 each) - \$70 (assuming 2 batteries purchased).
- `pens`: Cost of pens (5 each) - \$0 (assumed no pens purchased).
- `shipping`: Variable to store the determined shipping cost.
- `currency`: Currency symbol - \$ (USD).
- `location`: Customer's location (e.g., "RSA" for South Africa).
- `customers`: Number of customers for the order.

### ## Calculate Shipping

The script first checks the `location` variable to determine the shipping cost. If the location is "RSA," the shipping cost is set to `RSA\_SHIPPING`. If the location is "NAM," the shipping cost is set to `NAM\_SHIPPING`. If the location is "NK" (assumed banned country), it will log a warning message (`BANNED\_WARNING`).

## ## Calculate Total Cost

The total cost of all items is calculated by summing the individual item costs: `shoes + toys + shirts + batteries + pens`.

## ## Check Free Shipping Eligibility

If the total cost is equal to or exceeds \$1000 and the location is either "RSA" or "NAM," the script checks if `customers` is equal to 1. If true, it applies free shipping by setting `shipping` to 0. Otherwise, it logs a warning message (`FREE\_WARNING`).

## ## Display Total Cost with Shipping

Finally, if the `shipping` variable is not null (i.e., shipping is determined), it calculates the total cost with shipping and logs it as output with the currency symbol.

If the `shipping` variable is null (i.e., shipping is not determined), it logs "Shipping not determined."

---

2. Please show how you applied JSDoc Comments to a piece of your code.

```
/**
 * Calculates the square of a given number.
 * @param {number} num - The input number.
 * @returns {number} The square of the input number.
 */
function calculateSquare(num) {
  return num * num;
}

/**
 * Calculates the shipping cost based on the location.
 * @param {string} location - The location code (e.g., "RSA" for South Africa).
 * @returns {number|null} The shipping cost in USD or null if location is not supported.
 */
function calculateShippingCost(location) {
  const RSA_SHIPPING = 400;
```

```

const NAM_SHIPPING = 600;
const OTHER_SHIPPING = 800;

if (location === "RSA") {
    return RSA_SHIPPING;
} else if (location === "NAM") {
    return NAM_SHIPPING;
} else {
    // Return null for unsupported locations
    return null;
}
}

/**
 * Calculates the total cost of items and determines if free shipping applies.
 * @param {number} shoes - Cost of shoes in USD.
 * @param {number} toys - Cost of toys in USD.
 * @param {number} shirts - Cost of shirts in USD.
 * @param {number} batteries - Cost of batteries in USD.
 * @param {number} pens - Cost of pens in USD.
 * @param {string} location - The location code (e.g., "RSA" for South Africa).
 * @param {number} customers - Number of customers for the order.
 * @param {string} currency - Currency symbol (e.g., "$" for USD).
 */
function calculateTotalWithShipping(shoes, toys, shirts, batteries, pens, location,
customers, currency) {
    const FREE_WARNING = "Free shipping only applies to single customer orders";
    const BANNED_WARNING = "Unfortunately we do not ship to your country of
residence";

    // Determine shipping cost based on location
    const shipping = calculateShippingCost(location);

    // Determine total cost of items
    const totalCost = shoes + toys + shirts + batteries + pens;

    // Determine if free shipping applies
    if (totalCost >= 1000 && (location === "RSA" || location === "NAM")) {
        if (customers === 1) {

```

```

        shipping = 0;
    } else {
        console.log(FREE_WARNING);
    }
}

// Display total cost with shipping (if applicable)
if (shipping !== null) {
    const totalWithShipping = totalCost + shipping;
    console.log("Price:", currency + totalWithShipping);
} else {
    console.log("Shipping not determined.");
}
}

```

```

// Example usage
const shoes = 300;
const toys = 100 * 5;
const shirts = 150 * 0;
const batteries = 35 * 2;
const pens = 5 * 0;
const location = "RSA";
const customers = 1;
const currency = "$";

```

```

calculateTotalWithShipping(shoes, toys, shirts, batteries, pens, location, customers,
currency);

```

---

3. Please show how you applied the @ts-check annotation to a piece of your code.

```

// @ts-check

/**
 * Calculates the square of a given number.
 * @param {number} num - The input number.
 * @returns {number} The square of the input number.
 */
function calculateSquare(num) {

```

```

    return num * num;
}

/**
 * Calculates the shipping cost based on the location.
 * @param {string} location - The location code (e.g., "RSA" for South Africa).
 * @returns {number|null} The shipping cost in USD or null if location is not supported.
 */
function calculateShippingCost(location) {
    const RSA_SHIPPING = 400;
    const NAM_SHIPPING = 600;
    const OTHER_SHIPPING = 800;

    if (location === "RSA") {
        return RSA_SHIPPING;
    } else if (location === "NAM") {
        return NAM_SHIPPING;
    } else {
        // Return null for unsupported locations
        return null;
    }
}

/**
 * Calculates the total cost of items and determines if free shipping applies.
 * @param {number} shoes - Cost of shoes in USD.
 * @param {number} toys - Cost of toys in USD.
 * @param {number} shirts - Cost of shirts in USD.
 * @param {number} batteries - Cost of batteries in USD.
 * @param {number} pens - Cost of pens in USD.
 * @param {string} location - The location code (e.g., "RSA" for South Africa).
 * @param {number} customers - Number of customers for the order.
 * @param {string} currency - Currency symbol (e.g., "$" for USD).
 */
function calculateTotalWithShipping(shoes, toys, shirts, batteries, pens, location,
customers, currency) {
    const FREE_WARNING = "Free shipping only applies to single customer orders";
    const BANNED_WARNING = "Unfortunately we do not ship to your country of
residence";

```

```
// Determine shipping cost based on location
let shipping = calculateShippingCost(location);

// Determine total cost of items
const totalCost = shoes + toys + shirts + batteries + pens;

// Determine if free shipping applies
if (totalCost >= 1000 && (location === "RSA" || location === "NAM")) {
  if (customers === 1) {
    shipping = 0;
  } else {
    console.log(FREE_WARNING);
  }
}

// Display total cost with shipping (if applicable)
if (shipping !== null) {
  const totalWithShipping = totalCost + shipping;
  console.log("Price:", currency + totalWithShipping);
} else {
  console.log("Shipping not determined.");
}

// Example usage
const shoes = 300;
const toys = 100 * 5;
const shirts = 150 * 0;
const batteries = 35 * 2;
const pens = 5 * 0;
const location = "RSA";
const customers = 1;
const currency = "$";

calculateTotalWithShipping(shoes, toys, shirts, batteries, pens, location, customers,
currency);
```

---

4. As a BONUS, please show how you applied any other concept covered in the 'Documentation' module.

The `@typedef` tag is useful for documenting custom types, particularly if you wish to refer to them repeatedly. These types can then be used within other tags expecting a type, such as `@type` or `@param`.

Use the `@callback` tag to document the type of callback functions.

```
/**
 * @typedef {Object} Item
 * @property {number} shoes - Cost of shoes in USD.
 * @property {number} toys - Cost of toys in USD.
 * @property {number} shirts - Cost of shirts in USD.
 * @property {number} batteries - Cost of batteries in USD.
 * @property {number} pens - Cost of pens in USD.
 */

/**
 * @typedef {Object} ShippingInfo
 * @property {string} location - The location code (e.g., "RSA" for South Africa).
 * @property {number} customers - Number of customers for the order.
 * @property {string} currency - Currency symbol (e.g., "$" for USD).
 */

/**
 * Calculates the shipping cost based on the location.
 * @param {string} location - The location code (e.g., "RSA" for South Africa).
 * @returns {number|null} The shipping cost in USD or null if location is not supported.
 */
function calculateShippingCost(location) {
  const RSA_SHIPPING = 400;
  const NAM_SHIPPING = 600;
  const OTHER_SHIPPING = 800;

  if (location === "RSA") {
    return RSA_SHIPPING;
  }
}
```

```

    } else if (location === "NAM") {
        return NAM_SHIPPING;
    } else {
        // Return null for unsupported locations
        return null;
    }
}

/**
 * Calculates the total cost of items and determines if free shipping applies.
 * @param {Item} items - Object containing the cost of individual items.
 * @param {ShippingInfo} shippingInfo - Object containing shipping and customer
information.
 */
function calculateTotalWithShipping(items, shippingInfo) {
    const FREE_WARNING = "Free shipping only applies to single customer orders";
    const BANNED_WARNING = "Unfortunately we do not ship to your country of
residence";

    // Determine shipping cost based on location
    let shipping = calculateShippingCost(shippingInfo.location);

    // Determine total cost of items
    const totalCost = items.shoes + items.toys + items.shirts + items.batteries +
items.pens;

    // Determine if free shipping applies
    if (totalCost >= 1000 && (shippingInfo.location === "RSA" || shippingInfo.location ===
"NAM")) {
        if (shippingInfo.customers === 1) {
            shipping = 0;
        } else {
            console.log(FREE_WARNING);
        }
    }
}

// Display total cost with shipping (if applicable)
if (shipping !== null) {
    const totalWithShipping = totalCost + shipping;

```



```
    console.log("Price:", shippingInfo.currency + totalWithShipping);  
  } else {  
    console.log("Shipping not determined.");  
  }  
}
```

// Example usage

```
const items = {  
  shoes: 300,  
  toys: 100 * 5,  
  shirts: 150 * 0,  
  batteries: 35 * 2,  
  pens: 5 * 0,  
};
```

```
const shippingInfo = {  
  location: "RSA",  
  customers: 1,  
  currency: "$",  
};
```

```
calculateTotalWithShipping(items, shippingInfo);
```

---