

# OIM3690 - Web Technologies



# **JavaScript - Form Validation**

# Processing Form Data

- Check if `input` is blank
  - Use `.value.length` – the number of characters typed in the form `input`
  - Example:

```
if (myForm["userName"].value.length == 0){  
    ...  
}  
// or if(myForm["userName"].value=="")
```

- Check if the entered value is a number
  - Use `isNaN()`
  - Example: `isNaN(myForm["age"].value)`
    - It returns `true` if the user entered a **non-numerical** value.
    - It returns `false` if the user typed a **numerical** value.

## Exercise: *ex17-1.html*

- Download *ex17-1.html* from [GitHub](#) (*OIM3690/resources/templates*)
- Create JavaScript functions
- When a user **clicks** the button, JavaScript will
  - capture the user inputs from the text fields and dropdown list
  - display all the inputs in `textarea`
  - check to make sure that the user **has entered something** in the text fields
  - check to make sure that the age value **is a numerical value**
- Can you **hide** age input when the `checkbox` (commented) is *checked*?

# Pair Programming

- **Driver**
  - Typing code
  - Sharing screen
- **Navigator**
  - Paying close attention to the code
  - Providing guidance and suggestions whenever possible
- Ideally, *Driver* and *Navigator* will switch roles

# Exercise: Celsius (C) - Fahrenheit (F) Converter

- Download **ex17-converter.html** from [GitHub](#) ([OIM3690/resources/templates](#))
- Write a function that converts one unit to the the other, and update the other input field accordingly
  - Formulae:
    - $F = C * (9/5) + 32$
    - $C = (F-32) * (5/9)$
- Validate user input to prevent unexpected behavior:
  - Display an error message if the user does not enter a value.
  - Display an error message if the user enters a non-numeric value.

# Pair Programming

1. Plan before coding (10 mins)
  - i. **DO NOT** start coding immediately.
  - ii. Write **pseudo-code** on a paper.
2. We will be back and summarize.
3. Code in VS Code (20 mins)
  - i. Copy **pseudo-code** into VS Code.
  - ii. Write JavaScript code based on **pseudo-code**.

# Benefits of Pair Programming

1. Constant feedback
2. Reduced frustration
3. Increased focus
4. Social interaction
5. Accountability
6. Collaborative skills
7. Real-world experience
8. Mentorship

source: [How Remote Pair Programming Works & Why it Can Change Your Life](#)



# Questions?

