JAVASCRIPT
FUNCTIONS

Subprograms

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

SUBPROGRAMS

- Modular Programming is a software design technique where you break each part of a program into individual separate modules.

- Subprograms (also know as procedures, methods subroutine and functions) are blocks of code set aside in a reusable bundle.

- You have already been using subprograms in this course. These are built in functions.

- There are three basic types of subprograms:
  1. Predefined
  2. Event driven (While these are still predefined, this can be considered a separate type of predefined)
  3. User Defined

Subprograms
predefined

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

SUBPROGRAMS – PREDEFINED

- Predefined subprograms are procedures that come with our language or from a library.

- Predefined subprograms are subprograms written by other programmers.

- Some examples are:
  - Join()
  - Substring()
  - ToString()

Subprograms
predefined events

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

SUBPROGRAMS – PREDEFINED EVENT

- Event subprograms are procedures that allow us to interact with the environment.

- These are common events:
  - Mouse
  - Keyboard
  - Window
  - Input

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

Subprograms
User Defined

Subprograms – User defined

- User defined subprograms are the procedures that we will be focusing on for this portion of the course.

- User defined subprograms at the base level are just code blocks we may want to repeat.

- Think about how tedious it would be if every time we wanted to push a new element into an array, we had to write the underline code.

Institute of
Technology
Development
of Canada

475 Granville Street, Vancouver, BC, V6C 1T1
Phone: +1(604)558-8727, +1(604)409-8200
Toll Free: +1(888) 880-4410
Fax: +1(888) 881-6545
Web: WWW.ITDCANADA.CA
Email: STUDYING@ITDCANADA.CA

Subprograms – User defined

- In some languages subprograms are broken up into two types; functions and subroutines

- Functions return values, subroutines execute processes. Think about a bank that returns money when you withdraw money (return) vs Checking your balance (execute a process)

- User defined subprograms in JavaScript are called Functions and they both can return and not return values

FUNCTIONS

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

FUNCTIONS

- **Functions** are programming tools that allow reuse blocks of code that may be repeated in out program.

```
function AddNumbersFrom1To100() {
    let sum = 0;
    for (let i = 0; i <= 100; i++) {
        sum += i;
    }
    console.log(sum)
}
AddNumbersFrom1To100();
```

I have made a function and can name it what ever I want

I have "Called" this function. And every time I write this it will be do the same thing

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

FUNCTIONS

- When we "Call" a function this means we are going to be running it on that specific line. This will print Hello, World twice.

```javascript
function helloWorld() {
  console.log("Hello, World");
}


helloWorld();
helloWorld();
```

JavaScript
Parameters And Arguments

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

JavaScript Functions

- What if we want to Change the "Hello, World" ?
- What is we want to use the same function to write "Hello, JavaScript!" and "Hello, Local municipality!"
- How do you think we can do that?

JavaScript Functions

- We can input different data into our function by defining parameters and inputting arguments.

- Parameters are variables we define so we can pass data into the function

- Arguments are the data we "pass" into the functions

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

JAVASCRIPT FUNCTIONS

```javascript
function helloWorld(greeting, place ) {
  console.log(`${greeting}, ${place}`);
}



helloWorld("Hello", "Home");
helloWorld("Goodbye", "City");
```

This is a parameter

These are arguments

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

JavaScript Functions

- Exercise 1:
- Make a function named Sum() that has two integer parameters, add the two integers together and print to the Console. "Call" it twice
- Input form load= 10, 5
- Output = "Sum arguments 10 + 5 = 15"
- Input = 55, 20
- Output = "Sum arguments 55 + 20 = 75"

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

JavaScript Functions

- Exercise 2:
- Make the product (multiplication), quotient(division) and difference (Subtraction) Subroutines just like sum
- Input = 10, 5
- Output = "Product arguments 10 * 5 = 50"
- Input = 55, 20
- Output = "Product arguments 55 * 20 = 1100"
- Input = 10, 5
- Output = "Quotient arguments 10 / 5 = 2"
- Input = 20, 55
- Output = "Quotient arguments 55 / 20 = 1100"
- Input = 10, 5
- Output = "Difference arguments 10 - 5 = 5"
- Input = 20, 55
- Output = "Difference arguments 55 - 20 = 2.75"

SUBROUTINES SCOPE

JavaScript scope

- What is scope?

- Scope can be thought of as what can be seen and accessed in specific locations.

- When we refer to scope in programming, we are referring to the level of access a variable or function has.

- Some levels of Scope:
  - Block
  - Function
  - Module
  - Global

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

JavaScript scope

- We can think about the level of access ability like Canadian politics.

- The Federal Government has influence over Provincial and municipal

- Provincial has influence over municipal but very little over Federal

- Municipal has influence over itself but has little say in Provincial or Federal affairs.

JavaScript scope

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1 (604)558-8727, +1 (604)409-8200
TOLL FREE: +1 (888) 880-4410
FAX: +1 (888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

JavaScript scope

- This is like scope. Global variable can influence everything which means that if we aren't careful, it can cause issues with what goes on locally. Local variables can influence very little outside of its on locality.

- Local variables are generally a better choice. We can define these with let and const

JavaScript scope

```
var global = "global"
let module = "module";
function Scope() {
  let functional = "functional";
  console.log(`${module}, ${functional}`);
  for (let block = 0; block < 10; block++) {
    console.log(block);
  }
}
```

This is a global scope It can be accessed anywhere

This is a Modular scope It can be accessed anywhere in the Module

This is functional scope. It cannot be accessed outside of the function it is in

This is a Block scope. It can only be accessed in the current block like the for loop

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

JavaScript scope

- In some circumstances it can be ok to use global scope, but 90% of the time its preferable to use a local variable. Here are some reasons

  - We prevent name conflict. You will notice that the for loops all come with index. This is a block variable so if there are no for loops inside our for loops, we can use the index name for all of our loops. In some languages we could be reassigning the variable which would cause even more issues.

  - It saves memory. When we use a variable in a function or block it releases the memory as soon as it is done. If we use a more global scope, it will keep the memory allocated till the program has stopped running .

SUBROUTINES STACK

SUBROUTINES - STACK

- What is stack?
- Stack is a place in RAM where data gets stored in contiguously. This means that we "stack" one piece of data on another.
- Think of a tennis ball tube or pringles.

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1 (604)558-8727, +1 (604)409-8200
TOLL FREE: +1 (888) 880-4410
FAX: +1 (888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

SUBROUTINES - STACK

- The first ball in is the last one out or the last one in is the first one out.

- (Last In First Out -> LIFO)

- This is stack memory. The first piece of data in is the last piece of data out.

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

Subroutines - Stack

- When we run a subroutine, it gets stored in stack memory, when the subroutine finishes, the memory is released.

- So, when you run a subprogram, it does take up memory, this can be useful. We can use a technique called recursion to take advantage of the data being stored, Which we will touch on later.

- It can also cause issues, and that is when we run out of stack space. This is called a Stack Overflow. It's like having too many tennis balls in a tube.

FUNCTIONS
RETURNS

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

JAVASCRIPT – RETURNS

- What is the difference between functions that returns and one that doesn't return?

- As we have seen so far, functions can execute a process, that is all. But what if we wanted to keep and use that value? Like a random number generator? we don't just generate a number then walk away, we keep the value

- How can use do that?

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

JavaScript – Returns

- Functions *return* values that we can use after our function has stopped.

```javascript
function Square(x){
  return x * x
}
let mySquaredNumber = Square(5)
console.log(mySquaredNumber)
```

Here is the function that returns x * x

This will pass 25 into our variable then we can print this in the console

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

JavaScript – Returns

- Make a one function called arithmetic.

- Create three parameters
  1. Operator
  2. Number1
  3. Number2

- Pass in (+ or - or * or /) as your string arguments and 2 numbers for integers, do the relevant arithmetic based on the operator input and return the value.

- Input = *, 5 , 10 --> Output = 50

- Input = +, 10, 55 --> Output = 65

- Input = -, 42, 2 --> Output = 40

- Input = /, 100, 20 --> Output = 5

Functions
Early Returns

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

FUNCTIONS– EARLY RETURNS

- When using functions, you can only return once. This means that once you leave the function you can't go back.

Functions– Early Returns

- The last return will never be reached because the water return we always be returned then the function will end be left

```
function Square(order) {
  if (order === "Coffee") {
    return "We have Coffee";
  } else if (order === "Tea") {
    return "We have Tea";
  } else {
    return "You get water";
  }
  return "This will never be reached";
}
let mySquaredNumber = Square("");
console.log(mySquaredNumber);
```

Functions
Exercises

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

FUNCTIONS

Exercise:

- Look back at the 100 random number array we made.

- Move the find largest, find smallest, sum, average, and the (even, odd, zero counter) into separate subprograms.

Functions

Exercise:

- Make a Mini-ATM.
- You will have four functions
  1. Bank_Balance
  2. Withdrawal
  3. Spending_Balance
  4. Spend
- Use A while loop to continue run, one prompt to choose withdrawal, Bank_Balance , Spending_Balance and Spend
- Withdrawal returns a value to spend
- I will show you a demo I have made

FUNCTIONS
DIFFERENT EXPRESSIONS

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

FUNCTIONS

- In JavaScript we express functions like:

```
function MyFunction() {}
const MySecondFunction = () => {};
```

- The const version has become more popular due to its scope

FUNCTIONS
CALL BACK

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

CALLBACK FUNCTIONS

- In JavaScript, a callback is a function passed into another function as an argument to be executed later.

- Any function that is passed as an argument is called a callback function.

```
function MyFunctions(x) {}
function CallBack(){


}

MyFunctions(CallBack());
```

CALLBACK FUNCTIONS

- Common places where you will see callback functions are:
  - Events
  - setInterval() and setTimeout()
- These are used to execute other functions after a certain laps of time.

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

FUNCTIONS
SET INTERVAL AND TIMEOUT

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

CALLBACK FUNCTIONS setInterval()

- Set interval is a built in function that allows us to set a time based interval between code execution. The code will run infinitely unless otherwise specified.

- setInterval(function, 2000) ;

- 2000 = 2000 milliseconds which is 2 seconds

- This function will execute every 2 seconds

- If you want it to stop you can use clearInterval()

Callback Functions setTimeout()

- setTimeout(function, 2000) ;
- This function will execute only once after 2 seconds

FUNCTIONS
Events

EVENTS

- ## BASIC EVENTS
  - CLICK
  - CONTEXTMENU
  - MOUSEOVER
  - MOUSEDOWN
  - MOUSEUP
  - KEYPRESS
  - SUBMIT

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA

EVENTS

- There are two ways to make an event
  1. On____ = () =>{}
  2. addEventListener("", ()=>{})
- Both methods are acceptable and have minor differences. I prefer to use addEventListener.

EVENTS – ON EVENTS

- The onclick, onmouseover, etc. methods types are commonly used, they have some limitations based on only one type of event can be added. This means you can't have two click events. But you can add them to the HTML directly

Events – onEventListener

- The on event listener can have multiple events of the same type. This means you can have two different click events on the same element.

- You cannot put it directly in the html

475 Granville Street, Vancouver, BC, V6C 1T1
Phone: +1(604)558-8727, +1(604)409-8200
Toll Free: +1(888) 880-4410
Fax: +1(888) 881-6545
Web: WWW.ITDCANADA.CA
Email: STUDYING@ITDCANADA.CA

Events

```
const button = document.querySelector("h2");
button.addEventListener("click", (event) => {
  console.log("event listener", event);
});
button.onmouseover = (event) => {
  console.log("onclick", event);
};
```

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE: +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX: +1(888) 881-6545
WEB: WWW.ITDCANADA.CA
EMAIL: STUDYING@ITDCANADA.CA