

---

# OCaml: The Learning Language

Presentation By: J. Delage, R. Dinero, W. Peyou, and B. Koenen

---

---

# Overview

1. Introduction to OCaml
2. Language Basics
3. Recursion in OCaml
4. Memory Management
5. OCaml in Practice
6. Questions and Answers



---

# History of OCAML

- Functional Programming Language
- French Institute for Research in Computer Science and Automation - 1996
  - designed as an industrial language for environments “where a single mistake can cost millions and speed matters”--OCAML.org
- Its simplicity made it a good teaching language
- Lightweight programming language
- Command line language
  - There is no IDE for OCAML



---

# Language Basics

(\* This is a function to get the volume with 3 ints \*)

```
let volume x y z =  
    x * y * z;;
```

(\* This is a function to find the average of 3 floats \*)

```
let average x y z =  
    (x +. y +. z) /. 3.0;;
```

```
volume 4 3 2 = 24
```

```
average (3.1 +. 1) 4.3 9.2 = 5.87
```

---



---

# Language Basics

- Casting
  - in order to cast a type to another type, there are different build-in functions to assist, such as:
    - `int_of_float`
    - `char_of_int`
    - `int_of_char`



Ex:

```
let inString = 10;;  
let newString = string_of_int inString;;  
(*newString will be set to "10"*)
```

---

# Language Basics

## Data Types and Type Binding

- Data types include: int, char, float, unit, bool, string, and user-defined types
- New data types can be created, similar to objects
  - `type student =`  
`{Name:string; Year:int; Major:string; GPA:float};;`



---

# Language Basics

- Type Inference - the type of a collection (say, a collection of integers) is parameterized by the type of its elements automatically
- Memory Management is Automatic
- Compiled language with a Top-level Interactive loop
  - All code is compiled rather than interpreted
  - The Loop reads in the line, evaluates, and finally executes the code.
  - The result is print back out



---

## OCAML If/While Statements

- If/while expressions are much like the SILLY language we worked on

- Ex: `let sample = 10;;`

`if sample > 5 then`

`print_string "Sample is Indeed`

Greater than \n”

`else`

`print_string "Sample is Less than`

\n”;;

- “Done” was previously left over from earlier versions and is still included in While/For loops, but not for If Statements
- 







---

## Lists

- Created with [ ]
  - Values separated by ;
    - `let list = [1; 2; 3];;`
  - Add to list with ::
    - `4 :: list;`
    - `[4; 1; 2; 3]`
  - `let m = 0 :: list;`
    - `m = [0; 4; 1; 2; 3]`
  - Append to a list
    - `list@[element]`
-

---

# OCAML Recursion

- Recursion in OCAML is a little special, as it is a deliberate learning method
- Recursion must be deliberately called so that the function knows to recursively call the function
- ex: 

```
let rec range a b =  
    if a > b then  
    []  
    else  
    a :: range (a + 1) b;;
```



---

# Memory Management

- Ocaml features a built-in garbage collection function
  - `Gc.full_major();;`
- Automatic Memory Reclamation/Garbage Collection
  - Uses Reference Counts to free memory
  - Also has 2 heaps
    - The minor heap for information that is quickly used and then removed
    - The Major heap that stores long-lasting data
    - Minor Heap objects → Major Heap



---

# OCAML in Practice

- Primarily a Teaching Language
  - Used at Rice University, Brown University, Caltech, Harvard, ect.
  - 35 Universities use OCaml as an introductory language

- Industry

- Jane Street Capital
  - Primary development language
  - Trading systems
  - Quantitative research
  - Systems software
  - Systems administration



---

# Question and Answer

---

---

# Sources

<https://www.ocamlpro.com/ocaml/>

<https://ocaml.org/>

<https://ocaml.org/learn/success.html>

<https://www.youtube.com/playlist?list=PLea0WJq13cnCef-3KSU3qWFge9OGUIKx1>

<https://www.youtube.com/watch?v=v1CmGbOGb2I>

<https://www.janestreet.com/what-we-do/>

<https://caml.inria.fr/pub/docs/oreilly-book/html/book-ora087.html>

<https://caml.inria.fr/pub/docs/fpcl/fpcl-06.pdf>

---

