

# ClojureCLR



Bangkok .NET Users Group

# Why ClojureCLR?

- Very simple syntax
- Great for concurrent systems
- Use existing .NET libraries
- Functional programming

# Why not ClojureCLR?

- Very different from C#
- Large Clojure community, but small ClojureCLR community

ClojureCLR is a modern Lisp dialect that is designed to be hosted, and includes simple concurrency constructs

# A Modern Lisp Dialect

- Clojure is a “dialect” of Lisp, invented in 2007
- ClojureCLR programs are nested list data-structures.
- The **first** list element is a function that is called on the **rest** of the list elements.

```
(+ 5 4)
```

```
(+ (* 5 2) 4)
```

```
(print "The answer is" (+ 30 12))
```

# A Modern Lisp Dialect

- Calling functions

ClojureCLR

(+ 5 4)

=> 9

(+ 5 4 2)

=> 11

C#

5 + 4

=> 9

5 + 4 + 2

=> 11

# A Modern Lisp Dialect

- Defining data

## ClojureCLR

```
(def name "Nok")  
  
(format "Hello %s!" name)  
=> "Hello Nok!"
```

## C#

```
var name = "Nok";  
  
String.Format("Hello {0}!",  
name);  
=> "Hello Nok!"
```

# A Modern Lisp Dialect

## - Defining functions

### ClojureCLR

```
(defn cube [n]  
  (* n n n))
```

```
(cube 4)  
=> 64
```

### C#

```
int cube(int n)  
{  
    return n * n * n;  
}
```

```
cube(4);  
=> 64
```



# A Modern Lisp Dialect

- Create functions that describe your domain  
(or find a library that already does!)

```
(select users  
  (where {:active true}))  
  (order :created)  
  (limit 5))
```

```
(defroutes app  
  (GET "/" [] "<h1>Hello World</h1>")  
  (GET "/echo/:text" [text] text)  
  (POST "/contact-form" [c] (insert-contact c)))
```

ClojureCLR is a modern Lisp dialect that is  
designed to be hosted, and includes  
simple concurrency constructs

# Designed to be Hosted

## JVM

- Java API
- Maven
- JARs

## CLR

- .NET API
- Nuget
- DLLs

## Web

- Browser API
- JS libraries

Clojure API

# Designed to be Hosted

- Native interoperability with your chosen VM

## .NET Interoperability

```
(def location (Path/GetFullPath "."))  
=> "C:\Users\will\project"
```

```
(.StartsWith location "C:")  
=> true
```

# Designed to be Hosted

- Native interoperability with your chosen VM

## JavaScript Interop

```
(let [target (js/jquery "body")  
      name (js/prompt "Please enter your name")  
      greeting (str "Hello, " name)]  
  (.text target greeting))
```

ClojureCLR is a modern Lisp dialect that is designed to be hosted, and includes simple concurrency constructs

# Concurrency Constructs

- **ref** - Atomically update multiple values in transactions

```
(def files (ref []))  
(def recycle-bin (ref []))  
  
(defn move-to-recycle-bin [file]  
  (dosync  
    (alter files remove #{file})  
    (alter recycle-bin conj file)))
```

# Concurrency Constructs

- **go** - Runs a code block asynchronously (a “goroutine”)
- **chan** - Creates a queue for communication between goroutines

```
(let [ch (chan)]  
  (go (while true  
        (println "Read: " (<! ch))))  
  (go (>! ch "hi")  
      (<! (timeout 5000))  
      (>! ch "there")))
```



ClojureCLR is a modern Lisp dialect that is designed to be hosted, and includes simple concurrency constructs

Walkthrough