

Emacs isn't a text editor, it's a lifestyle.

▶ All hail Emacs.

Emacs? That sounds like an Apple™ thing.

- ▶ Emacs stands for **E**ditor **MAC**ro**S**.
- ▶ Emacs started as some macros for TECO (**T**ext **E**ditor and **C**orrector)
- ▶ Created in 1976 by some people, including Guy Steele.

Some stuff about Emacs.

- ▶ Emacs is a graphical text editor
 - ▶ `-nw` flag in case you gotta use a terminal
- ▶ `C-h t` to start the tutorial
 - ▶ Next slide will explain notation

Shortcuts.

- ▶ `C-h t` to start the tutorial
- ▶ Shortcuts show sequences of key presses
 - ▶ The `C-` prefix represents the `control` key being held
 - ▶ The `M-` prefix represents the `alt` (meta) key being held
- ▶ Some more examples:
 - ▶ `C-p`, `C-n`, `C-f`, and `C-b` move the cursor thing up, down, rightward, and leftward respectively.
 - ▶ Of course, you can make some shortcuts of your own.

Code Blocks

```
(message "Hello, World!")
```

```
def foo (n) :  
  if n == 0 :  
    return 1  
  return n * foo(n-1)  
return foo(5)
```

Games

```
(chess) ; With AI
```

```
(chess '(nil)) ; No AI
```

```
(chess-tutorial)
```

Games

(slime-volleyball)

Games

There are a lot of games, ok?

XKCD

```
(xkcd-get 378)
```

```
(butterfly)
```

Calculator

```
(calc)
```

You can also just do a code block if you wanna.

```
return "The answer to life, the universe, and everything is
```

Tables

Table 1

Emacs Usage (hours/day)	"Coolness"
0	0
1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100
11	121
12	144
13	169
14	196

Tables as input to code

```
import functools
max_coolness = functools.reduce(lambda a,b: max(a,b[1]), ta
return max_coolness
```

Tables as output from code

```
import random
tab = []
for j in range(4) :
    tab.append([])
    for i in range(4) :
        tab[j].append(random.randint(0,9))
return tab
```

Elisp (1)

About time, eh?

```
(message "Hello, World!")
```

- ▶ "LISP" = "Lots of Irritating Superfluous Parentheses"
- ▶ "E" = "Emacs"
- ▶ M-x ielm to run elisp.
- ▶ You can also run a line of elisp with C-x C-e.

Elisp (2)

Functions

```
(defun call-it-cool (thing) (message "%s is cool." thing))  
(call-it-cool "Emacs")
```

Elisp (3)

Lists

```
(list 1 2 3  
      4 5 6  
      7 8 9)
```


Elisp (4)

```
(switch-to-buffer "*Messages*")  
  
(switch-to-buffer "*scratch*")  
(goto-char (point-min))  
(set-mark-command (point))  
(goto-char (point-max))  
(delete-active-region)
```

Elisp (5)

```
( + 1 2 3  
  ( * 2 2 )  
)
```

Elisp (6)

```
(car '(1 2 7 3))
```

```
(cdr '(1 2 7 3))
```

Elisp (7)

```
(defun fac (n) (if (= n 0)
                    1
                    (* n (fac (- n 1))))
  ))
(fac 5)
```

Elisp (8)

```
(defun reduce (f l v)
  (if l
      (reduce f (cdr l) (funcall f v (car l)))
      v
  )
)
(reduce '+ '(8 6 7 5 3 0 9) 0)
```

Elisp (9)

```
(seq-reduce '+ '(8 6 7 5 3 0 9) 0)
```

```
(mapcar '1+ [1 2 3])
```

Elisp (10)

```
(defun example-interactive-function (x y s)
  (interactive "nFirst Number: \nnSecond Number: \nMString\n"
    (message "%d %s" (+ x y) s))
  (call-interactively 'example-interactive-function))
```

Elisp (11)

Characters are like integers

```
(- ?C ?A)
```

```
?A
```

```
(kbd "C-x M-c M- ")
```


Elisp (12)

```
(list
  (integerp 17)
  (integerp '(1 2 3))
  (sequencep '(1 2 3))
  (sequencep 17)
)
```

Elisp (13)

```
(list
  (format "%c" (elt "Some text" 3))
  (elt '(42 65 100 2) 2)
)
```

Elisp (14)

```
(cons 1 (cons 2 (cons 3 (cons 4 nil))))
```

```
'(1 . (2 . (3 . (4 . nil))))
```

```
'(1 2 3 4)
```

Elisp (15)

```
(setq counter-thing 0)
(setq string-thing "Some numbers:")
(while (< counter-thing 10)
  (setq string-thing (format "%s %s" string-thing counter-thing))
  (setq counter-thing (+ counter-thing 1))
)
string-thing
```

Elisp (16)

```
(let* ((a 12)
       (b (* a 7))
       (c (/ b 2)))
      c
    )
```

Elisp (17)

```
(set-background-color "black")  
(set-foreground-color "green")  
  
(set-background-color "white")  
(set-foreground-color "black")  
  
(set-background-color "#272822") ; Theme's Default  
(set-foreground-color "white")
```

Nullam eu ante vel est convallis dignissim. Fusce suscipit, wisi nec facilisis facilisis, est dui fermentum leo, quis tempor ligula erat quis odio. Nunc porta vulputate tellus. Nunc rutrum turpis sed pede. Sed bibendum. Aliquam posuere. Nunc aliquet, augue nec adipiscing interdum, lacus tellus malesuada massa, quis varius mi purus non odio. Pellentesque condimentum, magna ut suscipit hendrerit, ipsum augue ornare nulla, non luctus diam neque sit amet urna. Curabitur vulputate vestibulum lorem. Fusce sagittis, libero non molestie mollis, magna orci ultrices dolor, at vulputate neque

Elisp (18)

```
(highlight-regexp "[AaEeIiOoUu]" "hi-pink") ;M-s h r  
(unhighlight-regexp "[AaEeIiOoUu]") ;M-s h u
```

Nullam eu ante vel est convallis dignissim. Fusce suscipit, wisi nec facilisis facilisis, est dui fermentum leo, quis tempor ligula erat quis odio. Nunc porta vulputate tellus. Nunc rutrum turpis sed pede. Sed bibendum. Aliquam posuere. Nunc aliquet, augue nec adipiscing interdum, lacus tellus malesuada massa, quis varius mi purus non odio. Pellentesque condimentum, magna ut suscipit hendrerit, ipsum augue ornare nulla, non luctus diam neque sit amet urna. Curabitur vulputate vestibulum lorem. Fusce sagittis, libero non molestie mollis, magna orci ultrices dolor, at vulputate neque nulla lacinia eros. Sed id ligula quis est convallis tempor. Curabitur lacinia pulvinar nibh. Nam a sapien.

Elisp (19)

(highlight-changes-mode)

Nullam eu ante vel est convallis dignissim. Fusce suscipit, wisi nec facilisis facilisis, est dui fermentum leo, quis tempor ligula erat quis odio. Nunc porta vulputate tellus. Nunc rutrum turpis sed pede. Sed bibendum. Aliquam posuere. Nunc aliquet, augue nec adipiscing interdum, lacus tellus malesuada massa, quis varius mi purus non odio. Pellentesque condimentum, magna ut suscipit hendrerit, ipsum augue ornare nulla, non luctus diam neque sit amet urna. Curabitur vulputate vestibulum lorem. Fusce sagittis, libero non molestie mollis, magna orci ultrices dolor, at vulputate neque nulla lacinia eros. Sed id ligula quis est convallis tempor. Curabitur lacinia pulvinar nibh. Nam a sapien.

Elisp (20)

I ran `lorem-ipsuM-insert-paragraphs` to get the Lorem Ipsum.

```
(search-forward ".")
```

```
(lorem-ipsuM-insert-paragraphs)
```

```
; Make sure to click somewhere after the dot.
```

```
; You wouldn't want lorem ipsum in your code, would you?
```

```
; Also, be able to edit.
```

-

Terminal Emulators in Emacs

```
(term "/bin/zsh")
```

EWV

(ewv "https://www.google.com/")

Some Org mode stuff I forgot about

Collapse with Tab

▶ How many are checked? [3/5]

- Check / Uncheck
- Check / Uncheck
- Check / Uncheck
- Check / Uncheck [1/2]
 - Check / Uncheck
 - Check / Uncheck
- Check / Uncheck

The End.

The End?

The End

I'll keep going, then?

```
(setq number (% (+ (% (random) 10) 10) 10))  
(defun guess-number (guess) (interactive "nGuess: ")  
(defun guess-number-hi (guess) (interactive "nToo High: ")  
(defun guess-number-lo (guess) (interactive "nToo Low: ")  
(call-interactively 'guess-number)
```

The End!