

Tokenizer

- The tokenizer will follow the FSM all the way, every state and every line leading from it is coded in:
 - For this Reason there are no, negative integers and 0006 is a Octa
 - Coding according to the FSM means it follows every state exactly, if it is not detected in the diagram it will not exist in the code

Problem characters:

- Some characters in the unix bash have been known to cause some problems because they are unix commands and they manipulate the strings entered.

These chars are:

- `
- ~
- !
- \$
- %

How I handle Malformed strings

- As soon as a malformed character is encountered the token is thrown out
- The code will print out the malformed char
- The char will also be printed in HEX
- Malformed chars are constituted by anything that does not fit the state
- The tokens are separated by spaces, the code looks for spaces to separate the tokens
- You can input a blank before a number, it will be printed as a blank along with the HEX value 0000
- States:
 - Int: Defined as 1-9
 - Only digits
 - No negatives
- Octal
 - Starts with 0
 - followed by 1-7
- Hex
 - Starts with 0x
 - Cannot be a exponent
- Exponent
 - starts with a integer
 - followed by either a digit or a decimal
 - then followed by e or E
- Zero
 - zero states is when there is only a zero
 - decides where things will go after a 0 has been detected
- Float
 - Anything that is a decimal
 - No negatives

