ITPro Today™ Rust Programming Language Cheat Sheet

Action	Command
	Strings
Define variable as string	let string: &'static str = "string value";
Print string value	println!("{}", string);
Concatenate strings	let string1 = "Ice"; let string2 = “cream”; let cat_strings = [string1, string2].join(“\n”);
	Variables
Set variable as integer	let x = 1;
Set variable as string	let x = "string value";
	Math

Add numbers	let x = 1;
	let y = 2;
	let sum = x + y;

Subtract numbers	let x = 1.0;
	let y = 2.0;
	let diff = x - y;
	
	Note: For multiplication and division in Rust, define floating point values (e.g., 1.0) rather than standard integers (e.g., 1) to ensure accurate results.

Multiply numbers	let x = 1.0;
	let y = 2.0;
	let prod = x * y

Divide numbers	let x = 1.0;
	let y = 2.0;
	let quot = x / y;

	Lists
Create an array	let arr = [1,2,3,4]; Note: You cannot create new values for an array in Rust, although you can modify existing values. To create a list that allows the addition (or removal) of values, use a mutable vector instead of an array.
Print values of array	println!("{:?}",arr);
Create vector	let vec = vec![1, 2, 3, 4];
Print values of vector	println!("{:?}",vec);

Add value to vector	let mut vec = vec![1, 2, 3,4];
	vec.push(5);
	
	Note: The vector must be declared as mutable; otherwise you can’t add values.

Sort vector values alphanumerically	let vec = vec![2, 1, 3, 4]; vec.sort(); // values are now 1, 2, 3, 4
	Files

Open a file in read-only mode	use std::fs::File;
	use std::io::Read;
	fn main() {
	  let mut file = File::open(“/some/file”);
	}

Open a file in read-write mode	use std::fs::File;
	use std::io::Write;
	fn main() {
	  let mut file = File::open(“/some/file”);
	}

Open a file in append mode	use std::fs::File;
	use std::io::Write;
	fn main() {
	  let mut file = File::options().append(true).open(“/some/file”);
	}

	Command Line Arguments

Read a command line argument	use std::env;
	
	fn main() {
	  let args: Vec<_> = env::args().collect();
	  }

	Searches and regexes
Search for value in string	let string = "ice cream"; println!(“{}”, str1.contains(“cream”));

Search for string in list	let vec = vec!["1","2"];
	if vec.contains(&”1”) {
	  println!(“yes”);
	}

	Conditionals

Create a for loop	let vec = &[1, 2, 3];
	for val in vec {
	  println!(“Value is {}.”, val);
	}

Create a while loop	let mut i = 0;
	while i < 10 {
	  println!(“i is still less than 10!”);
	  i = i + 1;
	}

Create if and elif statements	let i = 1;
	 if i < 100 {
	  print!(“value is less than 100”);
	 } else if i > 0 {
	  print!(“value is greater than 100”);
	}

	Functions

Define a function	fn fun() {
	let i = 1;
	println!(“{}”, i);
	}

Call a function	fn fun() {
	let i = 1;
	println!(“{}”, i);
	}
	
	fn main() {
	fun();
	}