SESSION 3

Project : Version Control System

Programming Club | 2024

OVERVIEW

Golang Basics

- Types , Variable and Function Declaration
- Loops and conditional statements
- Structs, Methods and Interfaces
- Standard and custom Packages

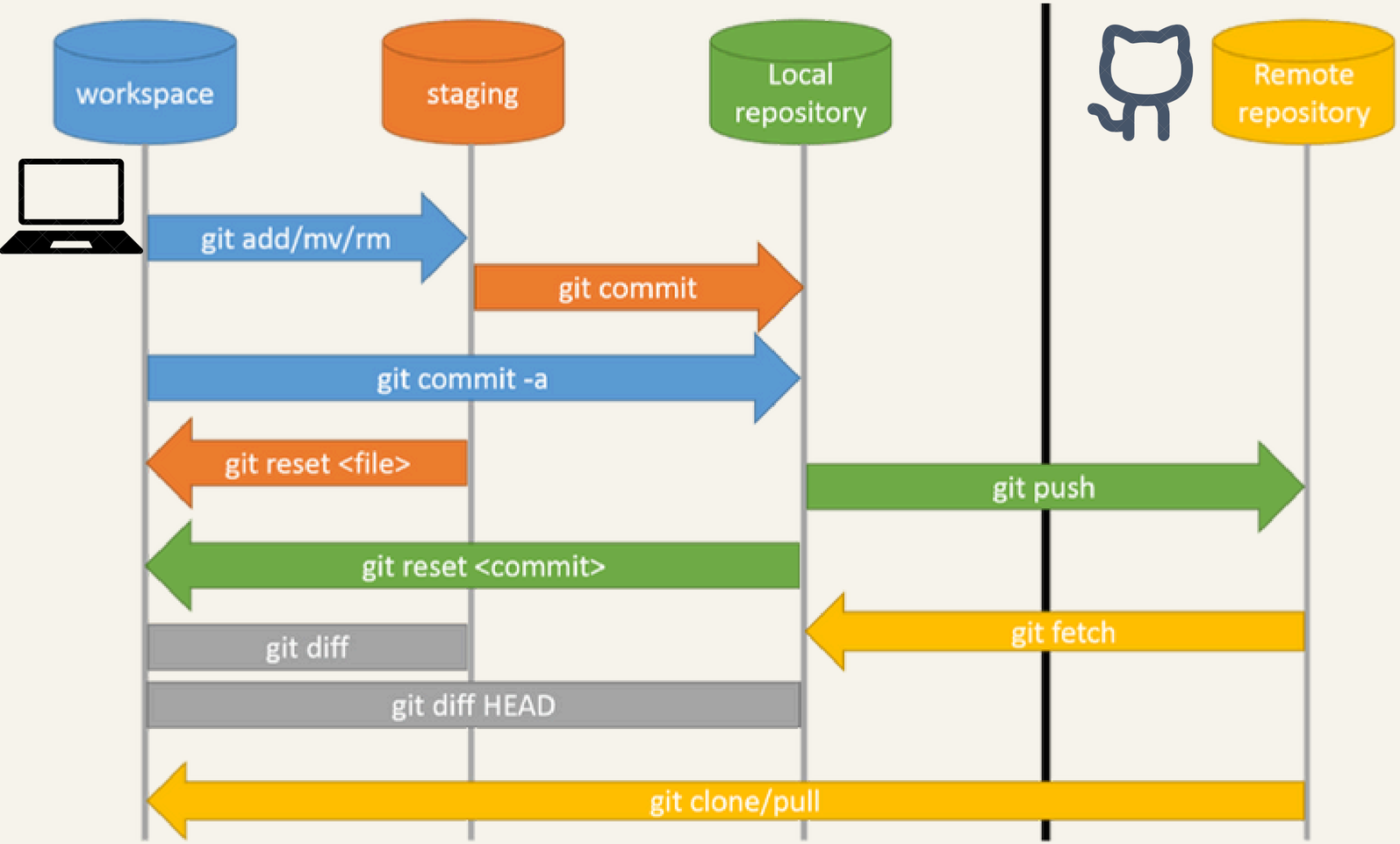
Git Basics

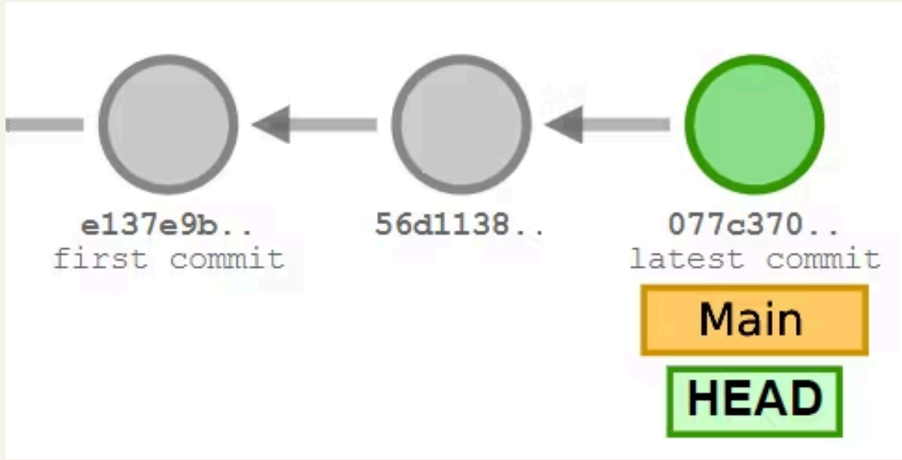
- Workspace, Staging Area, local repository and remote repository
- git init, git status, git add, git commit, git reset
- HEAD pointer and commit ID

Bash Basics

- ls, cd, mkdir, rm, touch , cat , echo
- options and flags (-l, -r)
- pipeline, alias

GIT BASICS RECAP





GIT BASICS RECAP

Root Commit

```
❯ cat Name.txt
My Name is Yash
❯ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will b
        Name.txt

nothing added to commit but untracked files present
❯ git add Name.txt
```

```
❯ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Name.txt

❯ git commit -m "First Commit"
[main (root-commit) 652b566] First Commit
 1 file changed, 1 insertion(+)
 create mode 100644 Name.txt
```

git log

```
commit 4cdf468e85a096d09f9f5305ece9df0f7a1d0a2e (HEAD → main)
Author: its-me-yps <yps8181@gmail.com>
Date:   Wed May 8 23:42:34 2024 +0530

    I lied

commit 652b566733cfac6729a03a33a62642dca7e38bde
Author: its-me-yps <yps8181@gmail.com>
Date:   Wed May 8 23:27:52 2024 +0530

    First Commit
~
```

git show

```
commit 4cdf468e85a096d09f9f5305ece9df0f7a1d0a2e (HEAD → main)
Author: its-me-yps <yps8181@gmail.com>
Date:   Wed May 8 23:42:34 2024 +0530

    I lied

diff --git a/Name.txt b/Name.txt
index b7e0d8e..691bc3f 100644
--- a/Name.txt
+++ b/Name.txt
@@ -1 +1 @@
-My Name is Yash
+My Name is not Yash, I was lying
(END)
```

“.GIT”

```
❯ git init
Initialized empty Git repository in /Users/yashpratapsingh/practice/.git/
❯ ls
❯ ls -a
.    ..    .git
❯ git status
On branch main

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

“.git” after git init

```
❯ tree .git
.git
├── HEAD
├── config
├── description
├── hooks
│   ├── applypatch-msg.sample
│   ├── commit-msg.sample
│   ├── fsmonitor-watchman.sample
│   ├── post-update.sample
│   ├── pre-applypatch.sample
│   ├── pre-commit.sample
│   ├── pre-merge-commit.sample
│   ├── pre-push.sample
│   ├── pre-rebase.sample
│   ├── pre-receive.sample
│   ├── prepare-commit-msg.sample
│   ├── push-to-checkout.sample
│   ├── sendemail-validate.sample
│   └── update.sample
├── info
│   └── exclude
├── objects
│   ├── info
│   └── pack
└── refs
    ├── heads
    └── tags

9 directories, 18 files
```

- Initializing git repository in a directory creates a “.git” directory
- “.git” stores the changes
- Analyzing “.git” directory will help us in understanding in internal working of git

- HEAD will store latest commit ID
- objects will store the changes committed in the repo

“.GIT”

```
❯ tree .git
.git
├── COMMIT_EDITMSG
├── HEAD
├── config
├── description
├── hooks
│   ├── applypatch-msg.sample
│   ├── commit-msg.sample
│   ├── fsmonitor-watchman.sample
│   ├── post-update.sample
│   ├── pre-applypatch.sample
│   ├── pre-commit.sample
│   ├── pre-merge-commit.sample
│   ├── pre-push.sample
│   ├── pre-rebase.sample
│   ├── pre-receive.sample
│   ├── prepare-commit-msg.sample
│   ├── push-to-checkout.sample
│   ├── sendemail-validate.sample
│   └── update.sample
├── index
├── info
│   └── exclude
└── logs
    ├── HEAD
    └── refs
        └── heads
            └── main
```

1

“.git” after staging two changes and committing them

New additions in “.git” after commit

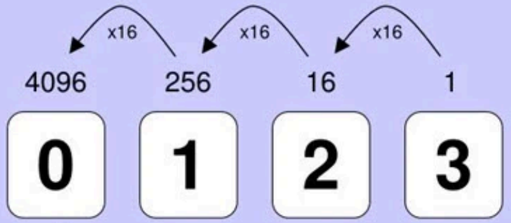
- COMMIT_EDITMSG
- index
- logs directory
- 4 new directories in objects with <u>hexadecimal</u> file name (two staged changes were committed)
- main in refs/heads

```
├── objects
│   ├── 25
│   │   └── 146458295609cd59c249545d8ba5af026ce036
│   ├── 3c
│   │   └── 99db513e6dc0a5a54f8b922e78de9455a9a02f
│   ├── 4b
│   │   └── 825dc642cb6eb9a060e54bf8d69288fbee4904
│   ├── 81
│   │   └── 154647fe9929f4ccfe6f726b66ce825579a221
│   ├── b7
│   │   └── e0d8ec04f00f11917393d6e73911d9d3c048f9
│   ├── info
│   └── pack
└── refs
    ├── heads
    │   └── main
    └── tags

17 directories, 28 files
```

2

HEXADECIMAL



As we move left, the column headings increase by a factor of **sixteen**

In each column we can have **sixteen** different digits

This number is:

1 x 256 + 2 x 16 + 3 x 1 = 291

*It's still **two hundred and ninety-one**, it's just written down differently*

- Base-16 number system
- 16 Unique symbols
- Numbers 0 to 9 and letters A to F
- Decimal 10 = A, 11 = B, 12 = C, 13 = D, 14 = E, 15 = F
- Can represent every byte (8 bits) as two hexadecimal digits
- Easier for humans to read hexadecimal rather than binary
- Hexadecimal numbers usually have either an 0x prefix or an h suffix.

Example: 0x3F7A, 85D2h

Number	0	1	2	3	4	5	6	7
Binary	0000	0001	0010	0011	0100	0101	0110	0111
Hexadecimal	0	1	2	3	4	5	6	7

Number	8	9	10	11	12	13	14	15
Binary	1000	1001	1010	1011	1100	1101	1110	1111
Hexadecimal	8	9	A	B	C	D	E	F

HEXADUMP

ASCII TABLE

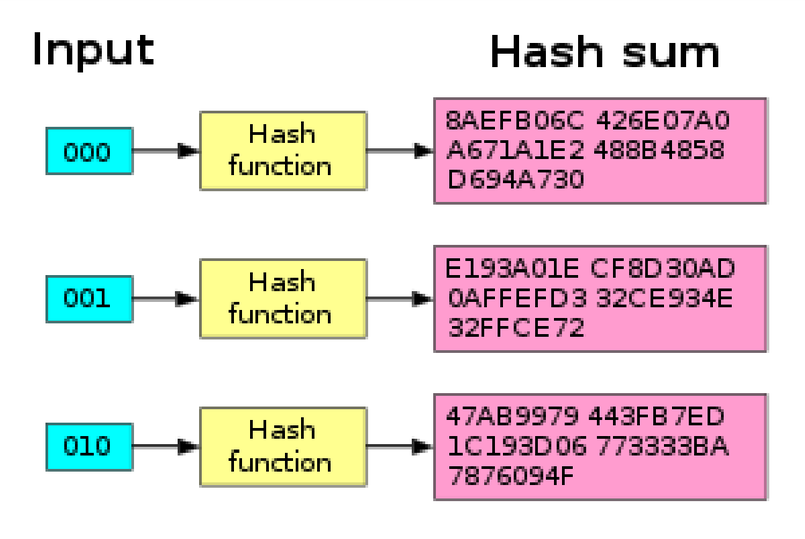
Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

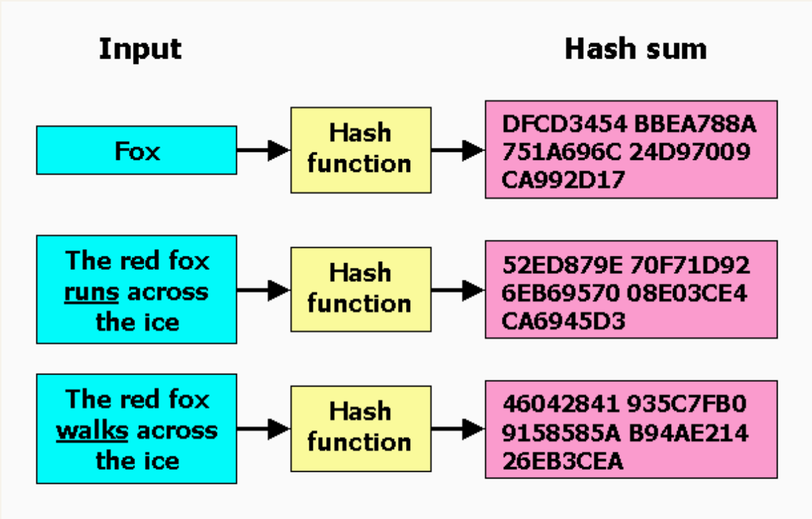
- Hexadecimal Representation of ASCII Characters

```
Offset    Hexadecimal                                     ASCII
00000000  00 60 94 A5 27 FF 00 04-AC EA 03 B9 08 00 45 00  .`..'.........E.
00000010  00 A3 04 03 40 00 40 06-B2 FA C0 A8 01 03 C0 A8  ....@.@.........
00000020  01 04 00 50 C0 5D B8 74-D9 30 9C C8 F3 12 50 18  ...P.].t.0....P.
00000030  83 2C A0 6D 00 00 48 54-54 50 2F 31 2E 30 20 32  .,.m..HTTP/1.0 2
00000040  30 30 20 4F 4B 0D 0A 53-65 72 76 65 72 3A 20 47  00 OK..Server: G
00000050  6F 53 65 72 76 65 2F 32-2E 35 30 0D 0A 43 6F 6E  oServe/2.50..Con
00000060  74 65 6E 74 2D 54 79 70-65 3A 20 74 65 78 74 2F  tent-Type: text/
00000070  68 74 6D 6C 0D 0A 43 6F-6E 74 65 6E 74 2D 4C 65  html..Content-Le
00000080  6E 67 74 68 3A 20 34 32-35 31 0D 0A 43 6F 6E 74  ngth: 4251..Cont
00000090  65 6E 74 2D 54 72 61 6E-73 66 65 72 2D 45 6E 63  ent-Transfer-Enc
000000A0  6F 64 69 6E 67 3A 20 62-69 6E 61 72 79 0D 0A 0D  oding: binary...
000000B0  0A 00 00 00 00                                  .....
```

- A hexdump is a tool that displays the contents of a file or data source in a readable format, hexadecimal and ASCII side by side
- Values in the central columns are the bytes of the file
- The leftmost column shows the total offset into the file that each row begins at, again in hexadecimal.

CRYPTOLOGICAL HASH

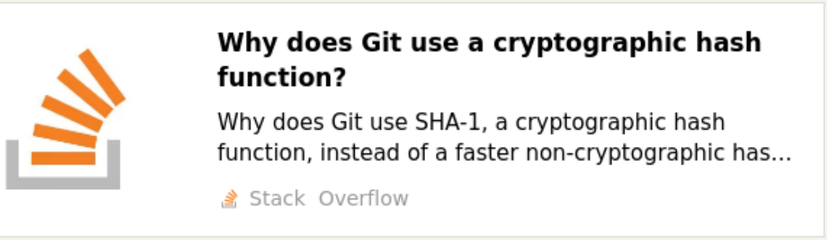


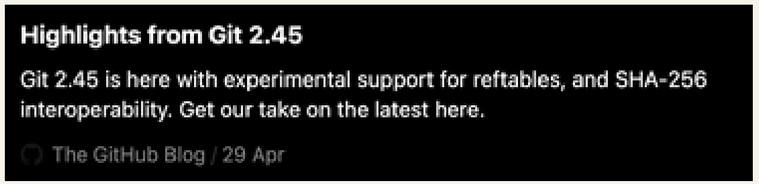


Properties of a Hash Function (H)

1. H can be applied to a block of data of any size
2. H produces a fixed-length output
3. $H(x)$ is relatively easy to compute for any given x
4. For any given value h, it is computationally infeasible to find x such that $H(x) = h$ (one-way)
5. For any given block x, it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$ (weak collision resistance)
6. It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$ (strong collision resistance)

CRYPTOLOGICAL HASH

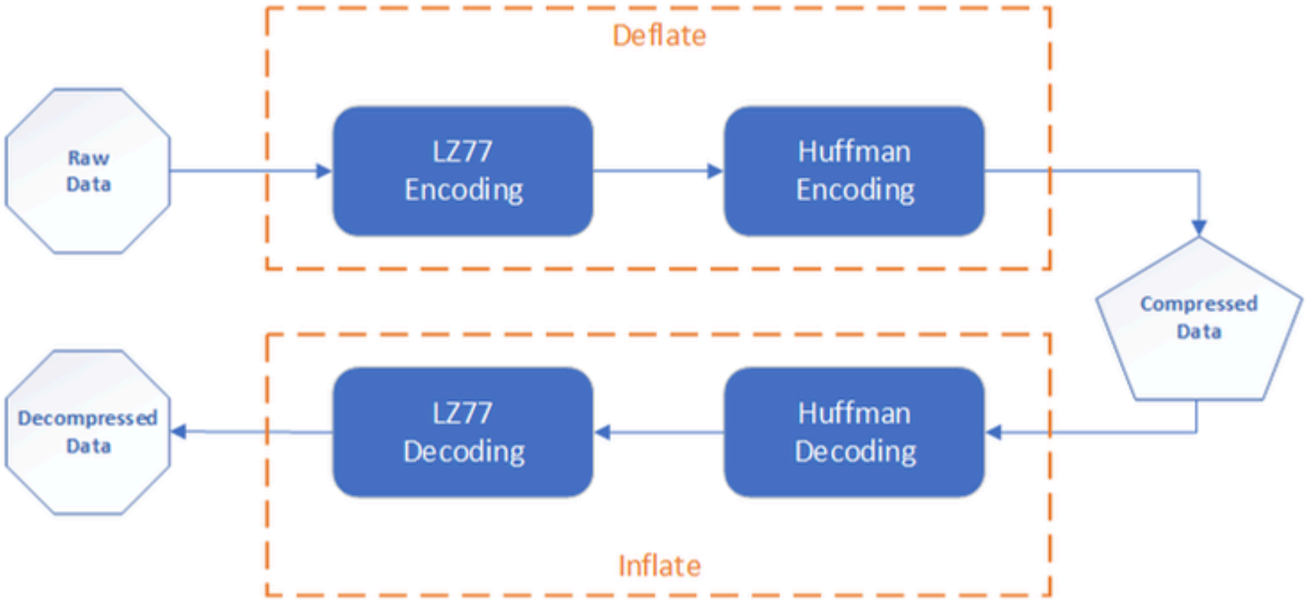




used in git

Name	Bits	Secure so far?
SHA256	256	Yes
SHA512	512	Yes
RIPEMD160	160	Yes
SHA-1	160	No. A collision has been found.
MD5	128	No. Collisions can be trivially created. The algorithm is also vulnerable to pre-image attacks, but not trivially.

ZLIB



zlib

- free, open-source software library and data format for lossless data compression and decompression

compress/zlib

- go's standard library implements zlib inflate and deflate in golang

Programming Club | 2024

THANK YOU

Project : Version Control System