Learn Python

2강 - 파이썬 자료형 & 제어문

1. 파이썬 자료형

1.1 파이썬 자료형 소개

- 자료형이란?
    - 프로그래밍을 할 때 쓰이는 숫자, 문자열 등 자료 형태로 사용하는 모든 것
    - 숫자형, 문자열, 리스트, 튜플, 딕셔너리, 집합

1.2 숫자형

- 숫자형
    - 숫자형은 정수, 실수, 8진수, 16진수 등의 숫자 형태를 의미한다.
    - 정수형
        - 2016, -100, 0 …
    - 실수형
        - 3.14, -5
    - 8진수와 16진수
        - 0o27 (8진수, 19)
        - 0xFF (16진수, 255)

1.3 문자열

- 문자열
    - 문자로 구성된 문자들의 집합이다
- “Hello, Python”
- “Hello, \”Python\””
- “(큰 따옴표)나 ‘(작은 따옴표)로 감싸서 표현한다.
- 이스케이프 코드
    - 미리 정의해 둔 "문자 조합”
    - \n(개행), \t(수평탭) 등

1.4 리스트

- 리스트
  - 자료형의 나열이다.
  - [(대괄호)로 감싸서 표현하고, ,(쉼표)로 구분한다.
    - list = [1, 2, 3, “가”, “나”, “다”]
    - 리스트의 첫 번째 인덱스는 0이다.
    - list[0] 은 1이다.
    - 리스트 안에 리스트를 포함할 수 있다.
    - list = [1, 2, [“가”, “나”]] >> list[2][0] 은 “가”이다.

1.4 리스트

- >>> list = [1,2,3,4,5]
- 리스트 슬라이싱
    - >>> list[0:2] 의 결과는 [1, 2]
    - >>> list[2:] 의 결과는 [3, 4, 5]
- 리스트 연산자
    - 리스트 더하기 (+)
        - >>> list + [6] 의 결과는 [1,2,3,4,5,6]
    - 리스트 반복하기 (*)
        - >> list * 2 의 결과는 [1,2,3,4,5,1,2,3,4,5]

1.4 리스트

- >>> list = [1,2,3,4,5]
- 리스트 수정, 변경, 삭제
    - >>> list[0] = 1000 의 결과는 [1000, 2, 3, 4, 5]
    - >>> list[1:2] = [“가”, “나”] 의 결과는 [1000, “가”, “나”, 2, 3, 4, 5]
    - >>> list[1] = [“다”] 의 결과는 [1000, [“다”], “가”, “나”, 2, 3, 4, 5]
- 리스트 삭제
    - >>> del list[1] 의 결과는 [1000,“가”, “나”, 2, 3, 4, 5]

1.4 리스트

- 리스트 함수
    - insert
        - 리스트에 값을 넣는다.
    - remove
        - 리스트의 값을 삭제한다.
- pop
    - 리스트의 값을 꺼낸다.
- count
    - 리스트의 개수를 구한다.
- extend
    - 리스트를 더한다.

1.5 튜플

- 리스트와 비슷하지만 ((괄호)로 둘러싼다.
- 값을 변경할 수 없다.
- 프로그램 동작 중 값을 변경시키지 않고자 한다면 튜플을 활용한다.

- tuple = (1, 2, 3)

1.6 딕셔너리

- key와 value로 이루어진 자료형
  - dic = {“name”: “raphael”, “age:” 30}
  - >>> dic[‘name’] 은 “raphael”
- 요소 추가와 삭제
  - dic[1] = 2 >>> key가 1이고 value가 2인 쌍을 추가한다.
  - del[1] >>> key가 1인 쌍을 삭제한다.
- 딕셔너리 keys()
  - >>> dic.keys() 수행하면 key 들의 리스트가 반환된다.

1.7 집합

- 중복을 허용하지 않는 집합 자료형
- >>> s1 = set([1,2,3])
- >>> s2 = set([3,4,5])
- 교집합(&)
    - >>> s1 & s2
- 합집합(|)
    - >>> s1 | s2
- 차집합(-)
    - >>> s1 - s2

1.8 변수

- 값을 저장하는 메모리의 위치를 가리키는 레퍼런스
    - a = 10
    - b = 20
    - c = 10
    - a, b, c = (10, 20, 10)
    - [a, b, c] = [10, 20, 10]
    - a = c = 10

1.8 변수

- 변수는 메모리의 위치를 의미한다.
  - >>> a = [1,2,3]
  - >>> b = a
  - >>> a[0] = 0
  - >>> b
  - 결과는 [0,1,2,3]

2. 파이썬 제어문

2.1 if문

- 만약 …이 참이라면 ()을 수행한다.

```
if 조건문:
    수행문1
else
    수행문2
```

```
>>> a = 1
>>> if a == 1:
        print(“참”)
    else
        print(“거짓”)
```

2.2 while문

- 반복해서 문장을 수행해야 하는 경우

```
while 조건문:
    수행문1
```

```
>>> a = 0
>>> if a > 10:
        a = a + 1
        print(“안녕”)
```

2.3 for문

- 가장 많이 활용하는 반복문

```
for 변수 in 리스트(튜플 or 문자열):
    수행문1
```

```
>>> a = [1, 2, 3]
>>> for i in a:
        print(i)
```

```
>>> a = [(1,2), (2,3), (3,4)]
>>> for (i, j) in a:
        print(i + j)
```

2.3 for문

```
>>> scores = [10, 100, 50, 70, 5]
>>> for score in scores:
        if score >= 60:
            print(“합격입니다.”)
```

```
>>> scores = [10, 100, 50, 70, 5]
>>> for score in scores:
        if score < 60:
            continue
        print(“합격입니다.”)
```

2.3 for문

- range(숫자) 함수는 0부터 숫자 미만까지의 숫자형 리스트를 만들어준다.
- range(숫자1, 숫자2) 함수는 숫자1부터 숫자2 미만까지의 숫자형 리스트를 만들어 준다.

```
>>> for i in range(0, 5) :
        if i == 2 :
            continue
        print( i )

>>> for i in range(0, 5) :
        if i == 2 :
            break;
        print( i )
```

2.4 함수

- 특정한 기능을 하는 프로그램의 일부분을 함수라고 한다.
- 여러번 호출되는 코드의 블락을 함수로 정의하고 활용한다.

```
def 함수이름(파라미터):
    수행문
    return 결과 >> 결과값이 있다면


def customPring(str):
    print(“프린트: ”+str)
```

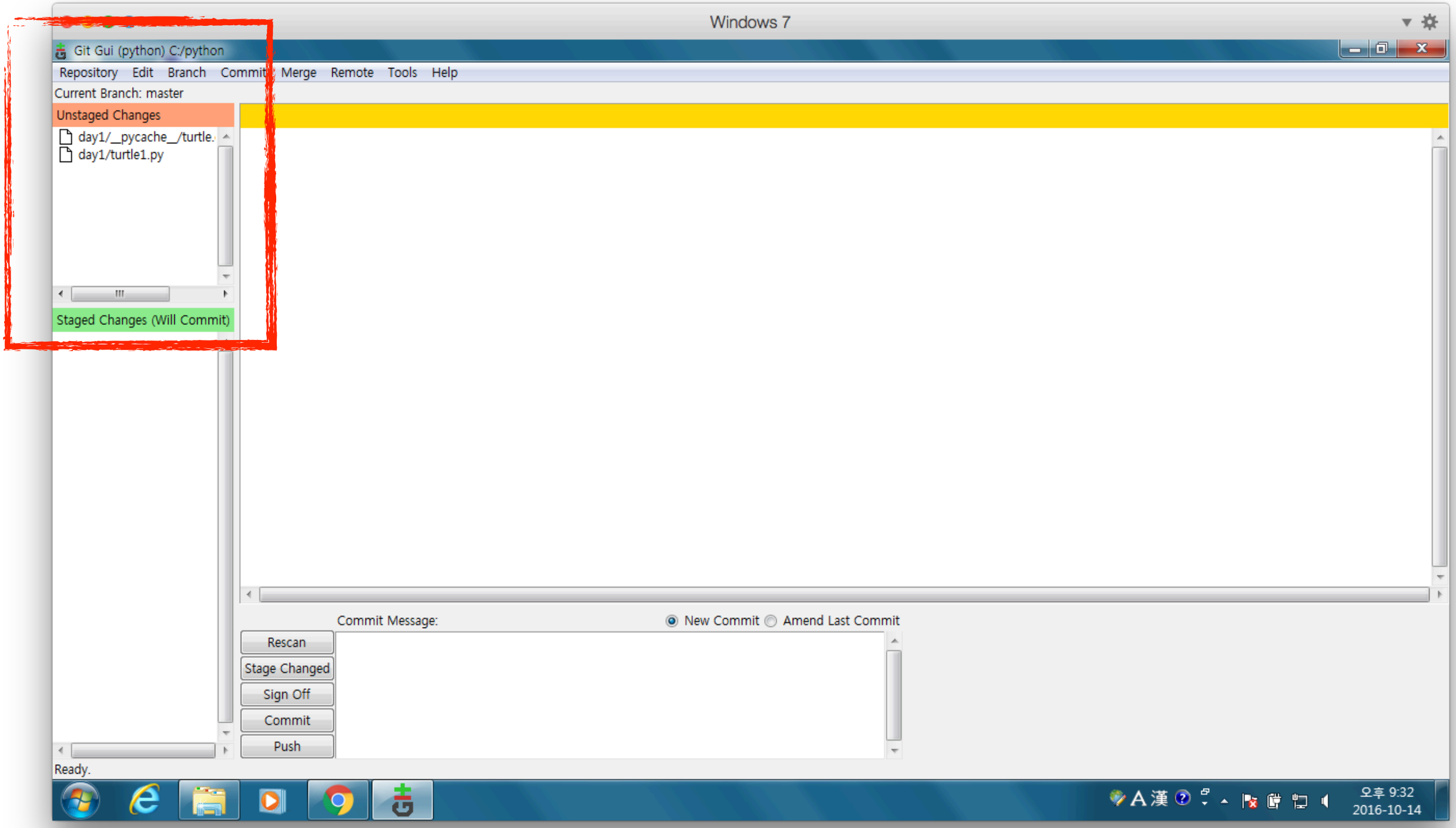
3. Git 간단 설명(1)

3.1 git clone

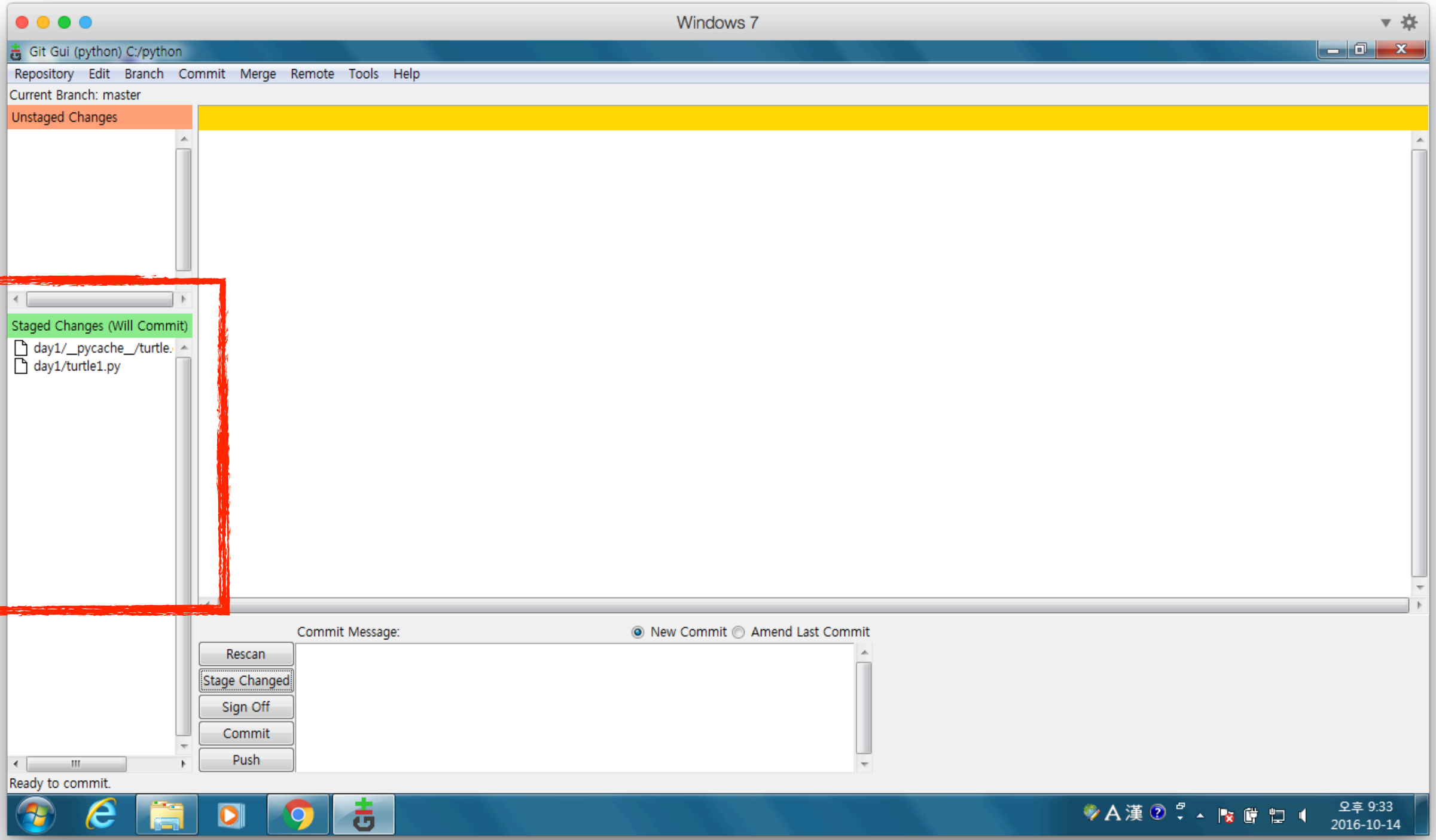
- 원격 저장소의 복제합니다.
- git clone /로컬/저장소/경로
- git clone 사용자명@호스트:/원격/저장소/경로

3.2 git 작업의 흐름

- 작업 디렉토리 - 인덱스 - HEAD
- 작업 디렉토리의 변경 내용은 unstaging area
- add를 해주면 인덱스 영역으로 staging area
- commit을 해주면 HEAD 영역
- 즉, 코드의 변경이 생시면 add를 한 뒤에 commit 한다.


Windows 7
Git Gui (python) C:/python
Repository Edit Branch Commit Merge Remote Tools Help
Current Branch: master
Unstaged Changes
day1/__pycache__/turtle.
day1/turtle1.py
Staged Changes (Will Commit)
Commit Message:
New Commit
Amend Last Commit
Rescan
Stage Changed
Sign Off
Commit
Push
Ready.
오후 9:32
2016-10-14

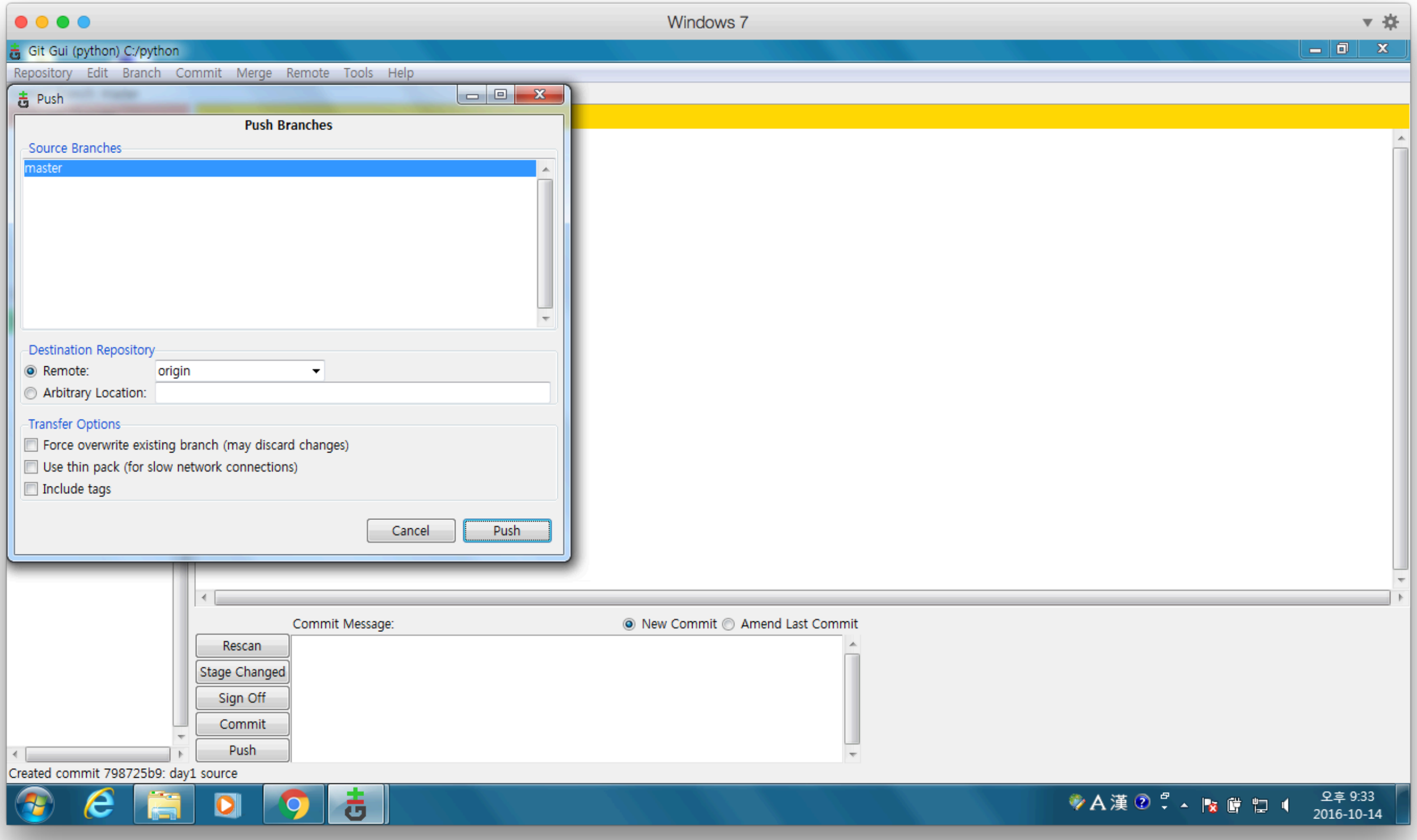
unstaged


Windows 7
Git Gui (python) C:/python
Repository Edit Branch Commit Merge Remote Tools Help
Current Branch: master
Unstaged Changes
Staged Changes (Will Commit)
day1/__pycache__/turtle.
day1/turtle1.py
Commit Message:
New Commit
Amend Last Commit
Rescan
Stage Changed
Sign Off
Commit
Push
Ready to commit.
오후 9:33
2016-10-14

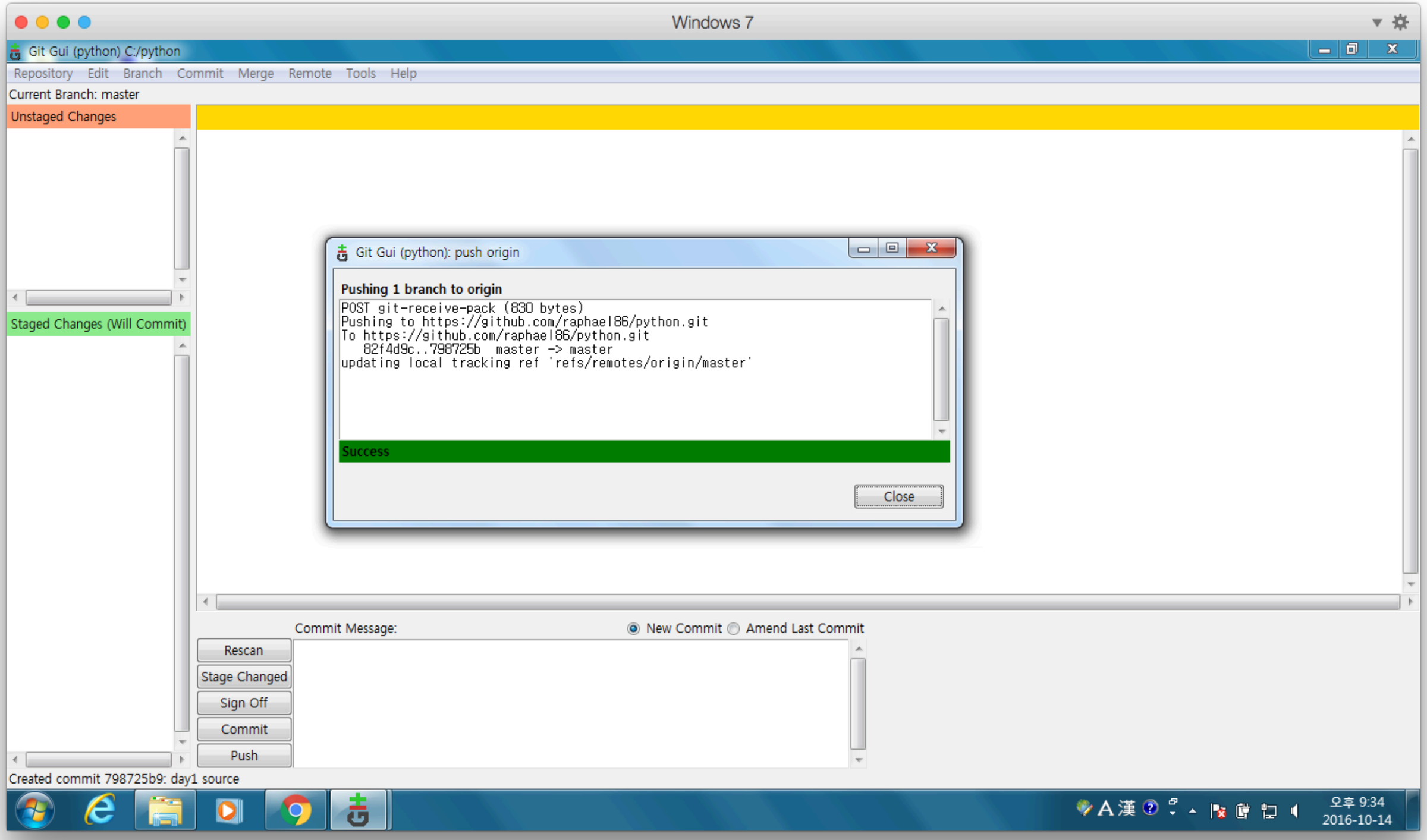
staged

3.3 git push

- commit한 내용을 원격 저장소로 올리는 작업
- git push (원격저장소) (브랜치)
- git push origin(깃헙저장소) master(기본 브랜치명)



origin master로 push 합니다.


Windows 7
Git Gui (python) C:/python
Repository Edit Branch Commit Merge Remote Tools Help
Current Branch: master
Unstaged Changes
Staged Changes (Will Commit)
Git Gui (python): push origin
Pushing 1 branch to origin
POST git-receive-pack (830 bytes)
Pushing to https://github.com/raphael86/python.git
To https://github.com/raphael86/python.git
82f4d9c..798725b master -> master
updating local tracking ref 'refs/remotes/origin/master'
Success
Close
Commit Message:
New Commit
Amend Last Commit
Rescan
Stage Changed
Sign Off
Commit
Push
Created commit 798725b9: day1 source
오후 9:34
2016-10-14

push 성공

4. 파이썬 코딩 실습

4.1 거북이 그래픽 활용

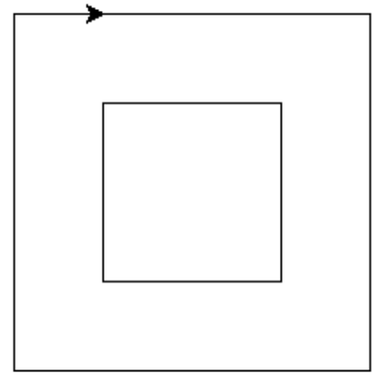
- turtle2_1.py
    - 반복문을 활용해서 사각형 2개 그려보기
- turtle2_2.py
    - 2배 크기의 사각형 n개 그리기

4.1 거북이 그래픽 활용

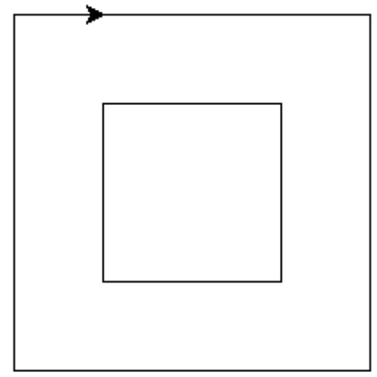
- turtle2_3.py
    - 반복문을 활용해서 n각형 그려보기
- turtle2_4.py
    - 반복문을 활용해서 원 선풍기 만들기

4.1 거북이 그래픽 활용

- turtle2_5.py
  - 반복문을 활용해서 사각형 2개 그려보기



- turtle2_6.py
  - 2배 크기의 사각형 n개 그리기



4.2 함수 응용

- function2_1.py
    - 사칙연산 함수 만들기
- turtle2_2.py
    - 다각형 그리는 함수 만들어보기