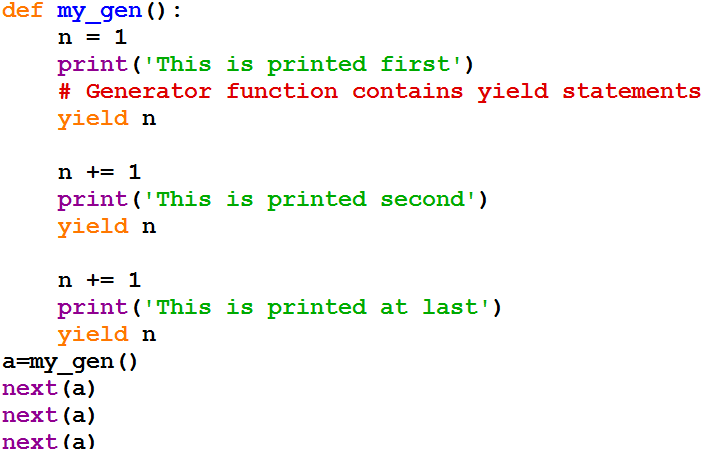
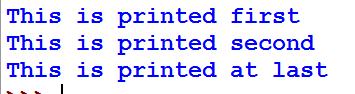
**Genarators í python**

Í einföldu máli þá er genarator fall sem getur skilað endurteknu gildi. Einu gildi í einu. Þegar búinn er til genarator fall er það gert eins og að skilgreina venjulegt fall nema hvað í stað þess að nota return er notað yield. Munurinn á þessu tvennu er sá að þegar notað er return í falli þá lokast fallið á meðan yield skipunin setur fallið í hvíldarstöðu. Næst þegar kallað er á fallið er haldið áfram frá því sem fyrr var horfið. Yield skipunin getur komið fyrir oftar en einu sinni. Skoðum dæmi:

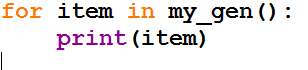


Úttak:

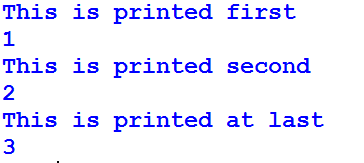


Hér sést vel hvernig staðan er geymd frá fyrri köllum (n er fyrst 1 svo 2 svo 3)

Í staðin fyrir að nota next() er líka hægt að keyra fallið í gegnum for slaufu því next() fall e r innbyggt í for slaufu. Dæmi



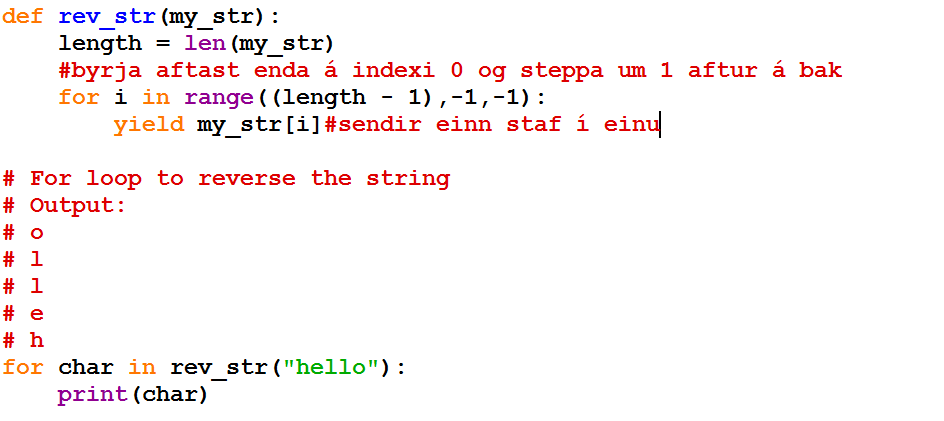
Úttak:



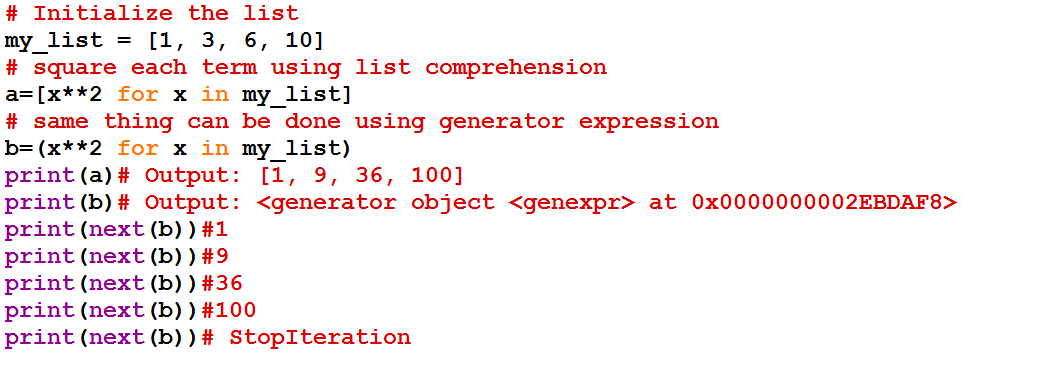
Þessi dæmi sýna aðeins hvernig genarators virka.

Skoðum alvöru dæmi um notkun á genarators.

Dæmi:



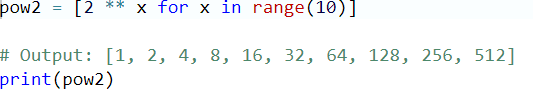
**Það er líka hægt að gera svona með genarators.**



Smá samantekt um vinnslu með lista.

## Góðar leiðir til að búa til nýja lista úr upprunalegum lista er að nota List Comprehension

**Dæmi 1: Býr til lista af tölum í öðruveldi frá 1 til 10**



**Dæmi 2: Býr til lista af tölum í öðruveldi frá 1 til 10. En bara þær tölur sem eru stærri en 5**



Úttak:



**Dæmi 3:Býr til lista af oddatölum á bilinu 1 til 20**



Úttak:



**Dæmi 4:Bræðir saman tvo strengjalista**



Úttak



**Æfingadæmi**

1. Hannið genarator fall sem skrifar á skjáinn í hvert skipti sem kallað er í það:
2. Nú hefur verið náð í mig í 1. sinn. Í næsta skiptið sem kallað er í fallið skrifast á skjáinn Nú hefur verið náð í mig í 2. sinn. Þetta á að vera hægt að endurtaka allt að 5 sinnum.
3. Hannið genarator fall sem getur skrifað út lista af tölum eina tölu í einu í hvert skipti sem kallað er á fallið.

## Búið til lista með því að nota List Comprehension. Listinn á að hafa allar tölur á bilinu 1 til 1000 sem 5 gengur upp í og eru oddatölur.