

Apunts Python - Conceptes Clau per l'Examen

NIVELL BÀSIC

1. Entrada/Sortida Bàsica

```
python
# Input amb missatge
quantitat = int(input("Entra quantitat: "))

# Loop per demanar múltiples valors
noms = []
for i in range(quantitat):
    nom = input(f"nom {i+1} ? ")
    noms.append(nom)

# Mostrar resultats
print("Els noms són:", " ".join(noms))
```

Elements clau:

- `input()` + `int()` per convertir
- `range(quantitat)` per repetir N vegades
- `f-string` per textos dinàmics

2. Control de Repetits amb Set

```
python
noms = set() # ★ Set evita repetits automàticament

while len(noms) < quantitat:
    nom = input(f"nom {len(noms)+1} ? ")
    if nom in noms: # ★ Comprovar si ja existeix
        print("Repetit, torna a intentar")
    else:
        noms.add(nom) # ★ Afegir al set
```

Elements clau:

- `set()` per elements únics
- `while` amb condició de longitud
- `in` per buscar en col·leccions

NIVELL MITJÀ

3. Dictionaris i Condicionals

python

```
persona = {
    "nom": input("Entra nom: "),
    "cognom": input("Entra cognom: "),
    "edat": int(input("Entra edat: ")),
    "pes": float(input("Entra pes: "))
}
```

★ Operador ternari

```
major = " (major d'edat)" if persona["edat"] >= 18 else ""
```

```
print(f'{persona["nom"]} {persona["cognom"]} {persona["edat"]} anys {persona["pes"]} Kg{major}')
```

Elements clau:

- Dictionaris amb claus string
- `valor if condició else valor` (ternari)
- Conversió a `int()` i `float()`

4. Llistes de Dictionaris i Càlculs

python

```
persones = [
    {"nom": "Bob", "cognom": "Esponja", "edat": 15, "pes": 0.2},
    {"nom": "Arenita", "cognom": "Mejillas", "edat": 19, "pes": 1.1}
]
```

★ Count amb condició

```
majors = sum(1 for p in persones if p["edat"] >= 18)
```

★ Mitjana

```
mitjana_pes = sum(p["pes"] for p in persones) / len(persones)
```

```
# ★ Noms repetits
noms = [p["nom"] for p in persones]
repetits = set(n for n in noms if noms.count(n) > 1)
```

Elements clau:

- `sum()` amb generator expression
- `set()` per trobar repetits
- Llistes per comprensió

NIVELL AVANÇAT

5. Filtres Complexos

```
python
# ★ Filtre múltiple
filtrades = [p for p in persones
             if p["edat"] >= 18 and p["pes"] >= mitjana_pes]
```

Elements clau:

- Llistes per comprensió amb condicions múltiples
 - Operadors lògics `and`
-

MANIPULACIÓ DE LLISTES

Transformacions

```
python
# ★ Valor absolut
llista2 = [abs(x) for x in llista1]

# ★ Filtrar parells
llista3 = [x for x in llista2 if x % 2 == 0]
```

Elements clau:

- `abs()` per valor absolut
- `% 2 == 0` per números parells

Eliminar Repetits

```
python
# ★ Convertir a set i tornar a llista
sense_repetits = list(set(llista))
```

Recorregut Invers

```
python
# ★ Reverse amb slicing
inversa = llista[::-1]
```

```
# ★ Amb reversed()
for element in reversed(llista):
    print(element)
```

Posicions Parells/Senars

```
python
# ★ Slicing amb salt
parells = llista[::2] # Posicions 0, 2, 4...
senars = llista[1::2] # Posicions 1, 3, 5...
```

GENERACIÓ DE LLISTES

Rangs Numèrics

```
python
# ★ range() amb list()
numeros_1_100 = list(range(1, 101))
numeros_101_200 = list(range(101, 201))
```

Elements clau:

- `range(inici, fi)` → fi NO inclòs
- `list()` per convertir range a llista

Validació d'Entrada

```
python
# ★ Loop infinit fins que valor sigui vàlid
while True:
    taula = int(input("Quina taula vols veure (1-10)?: "))
    if 1 <= taula <= 10:
        break
```

DICCIONARIS COMPLEXOS

Comptar Frequències

```
python
frequencia = {}
for lletra in frase:
    if lletra in "aeiou":
        frequencia[lletra] = frequencia.get(lletra, 0) + 1
```

Elements clau:

- `dict.get(clau, valor_per_defecte)`
- Condició `in` string

Posicions en Llistes

```
python
posicions = {}
for i, num in enumerate(llista):
    if num not in posicions:
        posicions[num] = []
    posicions[num].append(i)
```

Elements clau:

- `enumerate()` per índex i valor
- Llistes com a valors de diccionari

Combinacions

```
python
combinacions = {}
for dau1 in range(1, 7):
    for dau2 in range(1, 7):
        suma = dau1 + dau2
        if suma not in combinacions:
            combinacions[suma] = []
        combinacions[suma].append([dau1, dau2])
```

Elements clau:

- Nested loops
- Diccionaris amb llistes de llistes

Histograma Gràfic

```
python
# ★ Trobar mínim i màxim
minim, maxim = min(llista), max(llista)

for i in range(minim, maxim + 1):
    asteriscs = "*" * llista.count(i)
    print(f"{i:2} {asteriscs}")
```

Elements clau:

- `min()`/`max()` per trobar rangs
 - `*` per repetir strings
 - Format `{i:2}` per alineació
-

CONCEPTES DIFERENCIALS

★ `range()`

```
python
range(5)      # 0,1,2,3,4
range(1, 6)   # 1,2,3,4,5
range(1, 11, 2) # 1,3,5,7,9
```

★ Operador Ternari

```
python
resultat = "si" if condicio else "no"
```

★ Slicing

```
python
llista[inici:fi:pas]
llista[:2]  # Parells
llista[::-1] # Inversa
```

★ `enumerate()`

```
python
for i, valor in enumerate(llista):
    print(f"Posició {i}: {valor}")
```

★ `join()`

```
python
"".join(noms) # Uneix elements amb espai
```

★ Comprensió de Llistes

```
python
[x*2 for x in range(5) if x % 2 == 0]
```

A practicar: Cada concepte amb els exercicis corresponents! 🚀

map() - Resum Ràpid

Concepte Bàsic

```
python
map(funció, iterable)
```

- Aplica una funció a **cada element** d'una llista/iterable
- Retorna un **objecte map** (cal convertir amb `list()`)

Exemples Pràctics

Convertir tipus:

```
python
# Strings → Integers
nums_str = ["1", "2", "3"]
nums_int = list(map(int, nums_str)) # [1, 2, 3]
```

```
# Input múltiple
entrada = "10 20 30"
numeros = list(map(int, entrada.split())) # [10, 20, 30]
```