

=====

日商プログラミング検定 STANDARD (Python) 類似問題 セット1
解答・解説

=====

【知識科目】第1問(知識4択:20問)

1. 正解:② python

解説:Python の対話モード(インタラクティブシェル)はターミナルで「python」(環境によっては「python3」)と入力して起動する。
.exe は拡張子であり起動コマンドではなく、py.run や pyexec というコマンドは存在しない。

2. 正解:② 3

解説:「//」は切り捨て除算(floor division)を行う演算子。
 $7 \div 2 = 3.5$ だが、小数点以下を切り捨てて 3 になる。
「/」であれば 3.5(実数)になる。

3. 正解:③

解説:Python はブロックの区切りを波括弧やendではなく「インデント」(字下げ)で表す言語である。そのため同じブロック内の文は同じ深さのインデントでそろえる必要がある。そろっていないと IndentationError になる。

4. 正解:② **

解説:べき乗(累乗)は「**」演算子で表す。例: $2 ** 3$ は 8。
「^」はPythonではビット単位のXOR(排他的論理和)であり、べき乗ではない点に注意。

5. 正解:③ _total

解説:Python の変数名は英字またはアンダースコアで始まる必要があり、数字で始まる名前(2nd_value)は使用不可。また class は予約語(キーワード)であるため変数名にできない。ハイフンを含む my-value も使用できない(引き算と解釈されてしまう)。
アンダースコアで始まる _total は使用可能。

6. 正解:③ 6 Hi

解説:print(3 * 2, 'Hi') は、まず $3 * 2$ が計算されて 6 になり、print の複数引数はデフォルトで半角スペース区切りで表示される。
文字列はクォートを付けずに中身だけが表示されるため「6 Hi」となる。

7. 正解:① (1)と(2)

解説:Python の and/or は「真偽値」ではなく「最後に評価した値」を返す。

(1) $2 \text{ and } 0 \rightarrow 2$ は真なので0を返す $\rightarrow 0$

(2) $0 \text{ and } 2 \rightarrow 0$ は偽なので0を返す(0で評価終了) $\rightarrow 0$

(3) $2 \text{ or } 0 \rightarrow 2$ は真なので2を返す(0で評価終了) $\rightarrow 2$

(4) $0 \text{ or } 2 \rightarrow 0$ は偽なので2を返す $\rightarrow 2$
結果が0になるのは(1)と(2)。

8. 正解: ① $5 \& 2$

解説: ビット演算を2進数で考える。

$5 = 0101, 2 = 0010$

① $5 \& 2 = 0000 = 0$ (両方1のビットがない)

② $5 | 2 = 0111 = 7$

③ $5 \wedge 3 = 0101 \wedge 0011 = 0110 = 6$

④ $\sim 0 = -1$ (ビット反転、Pythonでは $\sim x = -x-1$)

結果が0になるのは①のみ。

9. 正解: ③ zero

解説: $x = 0$ のとき、 $x > 0$ は False、 $\text{elif } x < 0$ も False なので、
else節の `print('zero')` が実行される。

10. 正解: ③

解説: 仕様は「 $x \geq 10$ のときOKと表示し、それ以外(10未満)のときは何もしない」というもの。①はそのまま

「`if x >= 10: print('OK')`」と書かれており仕様通り。②④は

どちらも「10未満のときは何もしない(pass)、それ以外のときにOKと表示する」という書き方で、遠回りだが結果的に①と同じ動作になる。一方、③の「`if x < 10: print('OK')`」は x が10未満のときにOKと表示してしまい、本来の仕様(10以上でOK)と論理が逆転しているため誤り。

11. 正解: ③ 8

解説: `while...else` は、`while` がbreakされずに正常終了した場合にelse節が実行される。

$x=1 \rightarrow 2(1<5)$ 、 $2 \rightarrow 4(2<5)$ 、 $4 \rightarrow 8(4<5)$ 、8は5未満でないためループ終了、
breakしていないのでelse節が実行され `print(x)` $\rightarrow 8$ が表示される。

12. 正解: ① 2 5 8 (改行区切り)

解説: `range(2, 10, 3)` は 2, 5, 8 を生成する(10は含まれず、2から3ずつ増加)。`print(x)` は毎回改行するので、2, 5, 8がそれぞれ別の行に表示される。

13. 正解: ② [20, 30]

解説: スライス `L1[1:3]` はインデックス1から2まで(3は含まない)を取り出す。`L1 = [10, 20, 30, 40]` なので `L1[1]=20`, `L1[2]=30` で結果は [20, 30]。

14. 正解: ① [1, 2, 3, 1, 2, 3]

解説: リストに対する「*」演算子は繰り返し結合を意味する。

`[1, 2, 3] * 2` は要素を2回繰り返した `[1, 2, 3, 1, 2, 3]` になる
(数値のように各要素が2倍されるわけではない)。

15. 正解: ② `data.pop(0)`

解説: pop(インデックス) は指定した位置の要素を削除して返す。
pop(0) で先頭要素を削除できる。remove(0) は「値が0の要素」を
削除するメソッドであり、位置の指定ではない点に注意。

16. 正解: ② PJ(それぞれ改行)

解説: for lang in S の各要素 lang は文字列 'Python', 'Java'.
lang[0] はその文字列の先頭1文字を取り出すので、'P' と 'J' が
それぞれ別の行に表示される。

17. 正解: ② scores.values()

解説: 辞書から値(value)だけを取り出すメソッドは values()。
キーだけなら keys()、キーと値のペアなら items()。

18. 正解: ③ {1, 2, 3}

解説: set() は重複を取り除いた集合を作る。リスト[1,2,2,3,3,3]
から重複を除くと {1, 2, 3} になる(集合なので順序は保証されない)。

19. 正解: ③ 60

解説: dict.get(キー, デフォルト値) は、キーが存在すればその値を、
存在しなければ指定したデフォルト値を返す。'C' は d に存在しない
ため、デフォルト値の60が返る。

20. 正解: ③ len(data)

解説: リストの要素数(長さ)を求める組み込み関数は len()。
length() や size() という関数・メソッドは標準では存在しない。

【知識科目】第2問(穴埋め:2問)

【問題1】外れ値(重いデータ)の検出

【1】正解: (2) mean + 10

解説: 「平均より10以上大きいデータ」を判定する基準(しきい値)は
「平均値 + 10」。

【2】正解: (2) x >= threshold

解説: 「10以上大きい」は「差が10以上(=ちょうど10大きい場合も
含む)」という意味なので、境界値ちょうど10大きいデータも
「重いデータ」に含める必要がある。そのため、単なる
x > threshold(10より大きいデータのみ)ではなく、
x >= threshold(10以上大きいデータすべて)が正しい条件になる。

【問題2】ハスの葉の倍々増加

【1】正解: (2) 1

解説: 1日目の葉は1枚なので、初期値 leaves は1にする。

【2】正解: (3) leaves * 2

解説: 「前日の倍」に増えるので、leaves = leaves * 2 で更新する。

【3】正解: (2) n

解説: ループを抜けた時点の日数 n が「何日目を超えたか」の答えに

なる。leaves は枚数であり日数ではないため誤り。

【知識科目】第3問(読解:1問)

正解:① 84と36の最大公約数を求める。

解説:これはユークリッドの互除法のプログラムである。

a, b = 84, 36 から開始し、b が 0 になるまで「a, b = b, a % b」を繰り返す。この操作を繰り返すと最終的に a に最大公約数が残る。

実際に計算すると、 $84\%36=12 \rightarrow 36\%12=0$ となり、最大公約数は12になる。

【実技科目】

【問題1】正解:if score >= 60:

解説:60点以上で合格、それ以外は不合格という条件分岐なので、「score >= 60」を条件式にする。

【問題2】正解:total2 + abs(x - average)

解説:平均偏差は「各データと平均の差の絶対値」の合計を個数で割ったもの。差を絶対値にするために abs() を使い、total2 に加算していく。

【問題3】

【1】正解:age // 10 - 1

解説:年齢を10で割った商(10代なら1、20代なら2...)から1を引くと「10代=0、20代=1、30代=2...」という配列のインデックスに変換できる。

【2】正解:dosu[idx] + 1

解説:該当する年代の人数(度数)を1人分増やす(カウントアップ)。

【3】正解:i in range(5)

解説:dosu と labels の要素数(5区分)分だけ繰り返し、それぞれの人数を表示するため、i は 0~4 の範囲でループさせる。