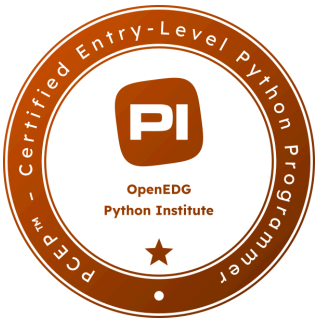
PCEP CERTIFICATION / WHAT HAS CHANGED

A clearer path to your PCEP™ certification



Your guide to PCEP-30-02 vs. PCEP-30-03

Same certification. New possibilities. A stronger foundation in Python.

PCEP-30-03 takes entry-level Python certification further with a broader, more comprehensive syllabus and a greater emphasis on practical programming skills. Discover what's changed, what remains essential, and how to build on what you already know to prepare confidently for the updated exam.

QUESTIONS	EXAM TIME	SECTIONS
30 → 35	**40 → 50 min**	**4 → 5**
Five additional questions	Ten more minutes	A dedicated fifth section

The big picture

Same credential. Updated skills.

Both versions lead to the *PCEP™ – Certified Entry-Level Python Programmer* certification. PCEP-30-03 retains the entry-level focus, while adding explicit coverage of practical tools such as sets, files, CSV data, and selected standard-library modules.

KEEP	ADD
The foundation	**More practical Python**
Variables, expressions, control flow, lists, tuples, dictionaries, strings, functions, and basic exceptions remain central.	PCEP-30-03 gives specific attention to sets, flexible function arguments, file operations, CSV, and module imports.

For candidates already preparing for PCEP-30-02: keep your existing fundamentals and use this guide to focus on the additional PCEP-30-03 objectives.

The exams at a glance

A quick side-by-side comparison of the exam experience.

EXAM DETAIL	PCEP-30-02	PCEP-30-03
Questions	30	35
Exam duration	40 minutes	50 minutes
Syllabus structure	4 blocks	5 sections
Passing score	70%	70%
Certification level	Entry	Entry
Prerequisites	None	None
Credential validity	Lifetime	5 years
Credential awarded	PCEP™	PCEP™

What this means

A broader exam, but a familiar entry-level goal

PCEP-30-03 introduces a more comprehensive syllabus, five additional questions, and ten more minutes of exam time. It places greater emphasis on practical Python programming while maintaining its focus on foundational skills rather than advanced software engineering. The 70% passing requirement remains unchanged.

The certification program has also adopted a **five-year credential validity period** to help ensure that newly earned certifications remain relevant as Python and modern programming practices continue to evolve.

What does not change

Your PCEP certification and previously earned credentials

Passing either exam version awards the same PCEP™ – Certified Entry-Level Python Programmer certification. The introduction of PCEP-30-03 does not invalidate previously earned credentials or require existing certification holders to retake the exam.

Credentials issued with lifetime validity remain valid for life. Credentials issued with a five-year validity period retain their original expiration dates. The updated validity policy does not apply retroactively to previously issued certifications.

Important dates

PCEP-30-02 will be retired on November 30, 2027

PCEP-30-02 will remain available during the transition period and will be officially **retired (sunsetted) on November 30, 2027**. After that date, candidates will no longer be able to take the PCEP-30-02 exam.

Candidates currently preparing for PCEP-30-02 can continue following its syllabus and take the exam before the retirement date. Those beginning their certification journey are encouraged to prepare for PCEP-30-03, which reflects the updated and expanded assessment scope.

The retirement of PCEP-30-02 does not affect the validity of certifications already earned.

Important for exam day

Check the exact exam code

Make sure your syllabus, learning resources, and preparation products correspond to the exam version you intend to take. If you have already purchased a voucher, confirm its exam version, validity, and redemption terms in your candidate account or with your voucher provider.

Planning to take PCEP-30-02? Make sure you complete your exam no later than November 30, 2027. Voucher expiration dates do not extend the availability of a retired exam version. Do not assume that exam versions or vouchers are interchangeable.

How the syllabus has changed

The original four-block structure has become a five-section structure.

PCEP-30-02 – OLD EXAM BLOCKS	PCEP-30-03 – NEW EXAM SECTIONS
Block 1: Computer Programming and Python Fundamentals 7 questions · 18%	**Section 1: Computer Programming and Python Fundamentals** 6 questions · 17%
Block 2: Control Flow – Conditional Blocks and Loops 8 questions · 29%	**Section 2: Control Flow: Conditional Blocks and Loops** 8 questions · 23%
Block 3: Data Collections – Tuples, Dictionaries, Lists, and Strings 7 questions · 25%	**Section 3: Data Collections: Lists, Tuples, Sets, Dictionaries, and Strings** 9 questions · 26%
Block 4: Functions and Exceptions 8 questions · 28%	**Section 4: Functions and Exceptions** 7 questions · 20%
–	**Section 5: File Handling and Module Imports** 5 questions · 14%
30 questions · 100%	35 questions · 100%

The weights above are the officially published scoring weights. They are not necessarily identical to each section's share of the question count, particularly for PCEP-30-02.

What stays familiar

Four foundations carry forward

- Python program structure, literals, variables, expressions, and console input/output.
- Conditional execution and loops, including tracing program flow.
- Lists, tuples, dictionaries, strings, indexing, slicing, and iteration.
- User-defined functions, parameters, return values, scope, and basic exception handling.

The most visible change

A dedicated fifth section

File Handling and Module Imports is now explicitly assessed in its own section (5 questions, 14%). You should be comfortable with text files, CSV files, importing modules, and selected *math* and *random* operations.

What to add to your study plan

These skills are new or more explicitly developed in the PCEP-30-03 assessment objectives. Use the self-check questions to identify what you already know and where you should focus your preparation.

New and newly introduced coverage

TOPIC	WHAT TO STUDY & SELF-CHECK
SETS	Create and modify sets; recognize unique values, membership checks, and the absence of positional indexing. **SELF-CHECK** Can you explain why a set is different from a list?
TEXT FILE HANDLING	Open, read, write, and append text files using `open()`, file modes (`r`, `w`, `a`), and the `with` statement. Recognize common file access errors. **SELF-CHECK** Can you read information from a text file and save a modified version?
CSV FILES	Understand CSV structure; use `csv.reader()` and `csv.writer()` to read, process, and write tabular data. **SELF-CHECK** Can you read a CSV file, extract a value from each row, and write the results?
MODULE IMPORTS	Understand why modules are useful, import existing code using `import`, and access objects through module names. **SELF-CHECK** Can you import a module and call one of its functions correctly?

MATH MODULE	Use `math.ceil()`, `math.floor()`, `math.trunc()`, and `math.sqrt()` for basic mathematical operations. **SELF-CHECK** Can you distinguish rounding upward, rounding downward, and truncating a number?
RANDOM MODULE	Use `random()`, `seed()`, `choice()`, and `sample()` to generate pseudorandom values and select elements. Understand repeatable results and unique sampling. **SELF-CHECK** Can you explain how `choice()` differs from `sample()` and why a seed is useful?
STRING PROCESSING	Use case changes, searching, splitting, joining, trimming, and replacing; recognize methods that return new strings. **SELF-CHECK** Can you transform a sentence and predict the result?
STRING FORMATTING	Build f-strings and recognize basic use of `format()` and `%`-style formatting. **SELF-CHECK** Can you produce a formatted message using a variable or expression?
FLEXIBLE FUNCTIONS	Work with `*args` and `**kwargs` and know that they collect positional and keyword arguments. Combine them with regular and default parameters. **SELF-CHECK** Can you tell whether a function receives a tuple or a dictionary?
LOOP EXPRESSIONS	Recognize the assignment expression operator `:=`, particularly in suitable `while`-loop patterns. **SELF-CHECK** Can you trace a loop that updates a variable inside its condition?

Retained skills and coverage

TOPIC	WHAT TO STUDY & SELF-CHECK
CONDITIONAL LOGIC	Trace nested and multi-branch `if`, `elif`, and `else` statements; recognize redundant, incorrectly ordered, and unreachable branches. **SELF-CHECK** Can you identify which branch executes and explain why others are skipped?
LOOP CONTROL	Use sentinel values, recognize infinite loops, and distinguish `break`, `continue`, and `pass`. **SELF-CHECK** Can you determine when a loop stops and which statements execute on each iteration?
RANGES AND NESTED LOOPS	Work with `range()` using start, stop, and step values, including negative steps. Trace nested loops and process multidimensional data. **SELF-CHECK** Can you predict the output of two nested loops using different ranges?
LISTS AND MULTIDIMENSIONAL DATA	Access, copy, modify, search, and sort lists; work with two- and three-dimensional nested lists using multiple indexes. **SELF-CHECK** Can you locate and change an element inside a nested list?
TUPLES	Create empty, single-element, and multi-element tuples; use indexing and slicing; understand immutability and nested collections. **SELF-CHECK** Can you explain why some operations work on lists but not tuples?
DICTIONARIES	Create, update, and remove key-value pairs; test for keys and iterate using `keys()`, `values()`, and `items()`. **SELF-CHECK** Can you predict what a dictionary contains after several updates and removals?
FUNCTION RETURN VALUES	Return single or multiple values; distinguish `return` from `print()` and recognize implicit `None` results.

	SELF-CHECK Can you determine exactly what a function returns, including when it has no explicit `return` statement?
FUNCTION ARGUMENTS	Combine positional, keyword, and default arguments; recognize invalid calls caused by missing, duplicated, or incorrectly ordered arguments. **SELF-CHECK** Can you identify why a function call succeeds or raises an error?
VARIABLE SCOPE	Distinguish local and global names, recognize variable shadowing, and understand the use and limitations of `global`. **SELF-CHECK** Can you determine which variable a function accesses or modifies?
EXCEPTION HANDLING	Recognize common built-in exceptions, including `ValueError`, `TypeError`, `ZeroDivisionError`, `IndexError`, `KeyError`, and `NameError`. Trace `try`/`except` statements with multiple exception branches. **SELF-CHECK** Can you identify the matching handler when an error occurs?
PROGRAM STRUCTURE AND ERRORS	Distinguish lexical, syntactic, and semantic errors; understand Python execution, identifiers, indentation, statements, and PEP 8 naming recommendations. **SELF-CHECK** Can you identify why a short program is invalid or produces an unexpected result?
CONSOLE INPUT AND TYPE CONVERSION	Combine `input()`, `print()`, `int()`, `float()`, and `str()` in complete programs; understand that `input()` returns a string. **SELF-CHECK** Can you trace a program that reads a value, converts it, performs a calculation, and displays the result?

Topics that are no longer explicitly assessed

PCEP-30-03 also changes the emphasis of several existing syllabus areas. Some topics specifically listed in PCEP-30-02 are no longer explicitly named in the updated assessment objectives.

TOPIC	WHAT HAS CHANGED
BITWISE OPERATIONS AND NUMERAL SYSTEMS	Detailed binary, octal, and hexadecimal numeral-system operations and bitwise operators are no longer explicitly listed.
SCIENTIFIC NOTATION	Scientific notation is no longer identified as a separate assessment requirement.
LOOP `else` CLAUSES	The `for-else` and `while-else` constructs are no longer explicitly listed.
LIST COMPREHENSIONS	List comprehensions are no longer specifically identified as an assessment requirement.
RECURSION AND GENERATORS	Recursion and generators are no longer explicitly included among the function objectives.
ADVANCED EXCEPTION CONCEPTS	Detailed exception-class hierarchies, exception propagation across function boundaries, and delegated exception handling are no longer explicitly emphasized.

Important: These topics remain valid Python concepts and may be useful for further learning. However, when preparing specifically for PCEP-30-03, prioritize the requirements explicitly listed in its current syllabus.

How to use this checklist

If you are transitioning from PCEP-30-02 preparation, begin with the new topics, particularly sets, file handling, CSV processing, module imports, and the math and random modules. Next, strengthen your knowledge of string formatting, flexible functions, loop expressions, and the more explicitly defined collection and control-flow operations.

You do not need to relearn every topic from the beginning. Build on your existing knowledge, practice the additional skills, and use the self-check questions to identify areas that need more attention.

Your goal is not simply to memorize more Python syntax. It is to become more confident in reading, understanding, tracing, and writing practical Python programs.

Reference syllabi

- PCEP-30-02 examination syllabus: https://pythoninstitute.org/pcep-30-02-exam-syllabus
- PCEP-30-03 examination syllabus: https://pythoninstitute.org/pcep-exam-syllabus

Which exam version should you choose?

Both exam versions lead to the same PCEP certification. The right choice depends on where you are in your preparation.

YOUR SITUATION	OUR RECOMMENDATION
I'm just starting to learn Python.	Prepare for PCEP-30-03. Its updated syllabus offers broader coverage of foundational and practical Python skills.
I've already been studying for PCEP-30-02.	Keep building on your existing knowledge. If you are nearly ready, you can take PCEP-30-02 before its retirement date. Otherwise, use this guide to bridge the gaps and transition to PCEP-30-03.
I've purchased a PCEP-30-02 voucher.	Check your voucher's exam version, expiration date, and redemption terms. Make sure you can complete the exam before PCEP-30-02 is retired.
I'm already PCEP certified.	Your existing PCEP certification remains valid according to its original terms, and retaking the exam is not required. However, we strongly recommend taking PCEP-30-03 to update and validate your knowledge against the expanded syllabus, strengthen your practical Python skills, and demonstrate your commitment to continuous learning and professional growth.

Our recommendation: If you are beginning your certification journey now, choose PCEP-30-03. If you have already invested substantial time preparing for PCEP-30-02, evaluate your readiness and timeline before switching.