Tabulka Datových Typů v Pythonu

Datový Typ	Popis	Příklad	Reálné Použití	Operace a Funkce
int (Celé číslo)	Reprezentuje celá čísla.	a = 42	Počet položek, indexy v seznamu, iterace.	+, -, *, /, //, %, **, abs(), divmod(), pow(), round(), int()
float (Desetinné číslo)	Reprezentuje čísla s desetinnou čárkou.	pi = 3.14	Výpočty s přesností na desetinná místa, fyzikální veličiny.	+, -, *, /, //, %, **, abs(), round(), float(), porovnání <, >
complex (Komplexní číslo)	Čísla s reálnou a imaginární částí.	z = 3 + 4j	Práce s vektory, matematické operace s imaginárními čísly.	.real, .imag, abs(), conjugate(), +, -, *, /, pow()
bool (Logická hodnota)	Reprezentuje True/False.	flag = True	Logické podmínky, kontrola pravdivosti.	and, or, not, ==, !=, <, >, is, is not, bool()
str (Řetězec)	Sekvence znaků v uvozovkách.	name = "Alice"	Textová data, zprávy, zpracování vstupu.	+, *, len(), upper(), lower(), strip(), replace(), find(), split(), join(), format(), f"{}", in, not in, str()
list (Seznam)	Kolekce prvků různých typů, měnitelná.	nums = [1, 2, 3]	Ukládání datových sad, iterace, fronty a zásobníky.	append(), insert(), remove(), pop(), extend(), sort(), reverse(), len(), in, not in, iterace for, +, *, list()
tuple (N-tice)	Neměnná kolekce prvků.	coords = (10, 20)	Ukládání pevných datových struktur, návrat více hodnot z funkcí.	len(), count(), index(), iterace for, in, not in, slicing ([:]), tuple()
set (Množina)	Neměnná kolekce unikátních hodnot, bez pořadí.	unique = {1, 2, 3}	Odstranění duplicit, matematické operace na množinách (průnik, sjednocení).	add(), remove(), discard(), union(), intersection(), difference(), issubset(), issuperset(), set()
dict (Slovník)	Kolekce klíč-hodnota.	data = {"key": "value"}	Ukládání párových dat, rychlé vyhledávání podle klíče, konfigurace.	keys(), values(), items(), get(), update(), pop(), clear(), setdefault(), in, not in, dict()
NoneType (None)	Reprezentuje prázdnou hodnotu (žádná data).	result = None	Signalizace prázdných nebo neexistujících hodnot, výchozí hodnoty parametrů funkcí.	Porovnání is, is not.

bytes (Bytový řetězec)	Sekvence bytů, neměnná.	data = b"hello"	Manipulace s binárními daty, síťová komunikace, práce se soubory.	len(), decode(), +, *, iterace for, slicing ([:]), in, not in, bytes()
bytearray (Bytový pole)	Sekvence bytů, měnitelná.	data = bytearray(5)	Práce s binárními daty, které lze měnit.	append(), insert(), remove(), pop(), extend(), slicing, iterace for, bytearray()
range	Sekvence čísel generovaná na požádání.	r = range(5)	Iterace přes čísla, generování číselných sekvencí.	in, not in, len(), slicing ([:]), iterace for, list(range(5))

Datový Typ	Operace/ Funkce	Popis	Příklad Kódu
int	+	Sčítání čísel.	result = 5 + 3 # Output: 8
	-	Odčítání čísel.	result = 10 - 7 # Output: 3
	*	Násobení čísel.	result = 6 * 7 # Output: 42
	/	Dělení s desetinným výsledkem.	result = 10 / 3 # Output: 3.333...
	//	Celé dělení (bez zbytku).	result = 10 // 3 # Output: 3
	%	Modulo, zbytek po dělení.	result = 10 % 3 # Output: 1
	**	Mocnina.	result = 2 ** 3 # Output: 8
	abs()	Absolutní hodnota.	result = abs(-5) # Output: 5
	divmod()	Vrací podíl a zbytek (tuple).	quotient, remainder = divmod(10, 3) # Output: (3, 1)
	pow()	Mocnina s možností zbytku.	result = pow(2, 3, 5) # Output: 3
float	round()	Zaokrouhlení na požadovaný počet desetinných míst.	result = round(3.14159, 2) # Output: 3.14
	float()	Převod na desetinné číslo.	result = float("3.14") # Output: 3.14
complex	.real	Vrací reálnou část čísla.	z = 3 + 4j; real_part = z.real # Output: 3.0
	.imag	Vrací imaginární část čísla.	z = 3 + 4j; imag_part = z.imag # Output: 4.0
	abs()	Velikost komplexního čísla (modul).	z = 3 + 4j; magnitude = abs(z) # Output: 5.0
bool	and, or, not	Logické operace.	result = True and False # Output: False
	Chyba:510	Porovnání hodnot.	result = (5 == 5) # Output: True
	is, is not	Kontrola identity objektů.	result = (a is b)
str	len()	Délka řetězce.	length = len("hello") # Output: 5
	upper()	Převod na velká písmena.	text = "hello".upper() # Output: 'HELLO'
	lower()	Převod na malá písmena.	text = "HELLO".lower() # Output: 'hello'
	strip()	Odstranění bílých znaků z kraje.	text = " hello ".strip() # Output: 'hello'
	replace()	Nahrazení části řetězce jiným textem.	text = "hello world".replace("world", "Python") # Output: 'hello Python'
	split()	Rozdělení řetězce na seznam podle zadaného oddělovače.	words = "a,b,c".split(",") # Output: ['a', 'b', 'c']

	join()	Spojení seznamu řetězců do jednoho řetězce.	result = ",".join(['a', 'b', 'c']) # Output: 'a,b,c'
list	append()	Přidání prvku na konec seznamu.	nums = [1, 2]; nums.append(3) # Output: [1, 2, 3]
	insert()	Vložení prvku na zadaný index.	nums = [1, 2]; nums.insert(1, 99) # Output: [1, 99, 2]
	pop()	Odstranění a vrácení posledního prvku (nebo podle indexu).	nums = [1, 2, 3]; nums.pop() # Output: 3, nums: [1, 2]
	remove()	Odstranění prvního výskytu hodnoty.	nums = [1, 2, 3, 2]; nums.remove(2) # Output: nums: [1, 3, 2]
	sort()	Seřazení seznamu.	nums = [3, 1, 2]; nums.sort() # Output: [1, 2, 3]
	reverse()	Otočení seznamu.	nums = [1, 2, 3]; nums.reverse() # Output: [3, 2, 1]
	extend()	Přidání více prvků najednou.	nums = [1, 2]; nums.extend([3, 4]) # Output: [1, 2, 3, 4]
dict	keys()	Vrací seznam klíčů.	data = {'a': 1, 'b': 2}; keys = data.keys() # Output: dict_keys(['a', 'b'])
	values()	Vrací seznam hodnot.	data = {'a': 1, 'b': 2}; values = data.values() # Output: dict_values([1, 2])
	items()	Vrací seznam klíč-hodnota párů.	data = {'a': 1}; pairs = data.items() # Output: dict_items([('a', 1)])
	get()	Vrací hodnotu podle klíče (nebo výchozí hodnotu).	data = {'a': 1}; value = data.get('b', 'default') # Output: 'default'
	update()	Aktualizace slovníku.	data = {'a': 1}; data.update({'b': 2}) # Output: {'a': 1, 'b': 2}
	clear()	Vymazání všech klíčů a hodnot ve slovníku.	d = {'a': 1, 'b': 2}; d.clear() # Output: {}
	pop()	Odstraní a vrátí hodnotu pro daný klíč.	d = {'a': 1, 'b': 2}; value = d.pop('a') # Output: 1, d: {'b': 2}
	setdefault()	Vrací hodnotu pro klíč, nebo nastaví výchozí hodnotu, pokud klíč neexistuje.	d = {'a': 1}; value = d.setdefault('b', 2) # Output: 2, d: {'a': 1, 'b': 2}
	in	Kontrola, zda klíč existuje ve slovníku.	d = {'a': 1}; result = 'a' in d # Output: True
	not in	Kontrola, zda klíč neexistuje ve slovníku.	d = {'a': 1}; result = 'b' not in d # Output: True
tuple	+	Sčítání n-tic.	t1 = (1, 2); t2 = (3, 4); result = t1 + t2 # Output: (1, 2, 3, 4)

	*	Opakování n-tice.	t = (1, 2); result = t * 2 # Output: (1, 2, 1, 2)
	len()	Délka n-tice.	t = (1, 2, 3); length = len(t) # Output: 3
	count()	Počet výskytů hodnoty v n-tici.	t = (1, 2, 2, 3); result = t.count(2) # Output: 2
	index()	Vrací první index hodnoty.	t = (1, 2, 3); index = t.index(2) # Output: 1
set	add()	Přidání jednoho prvku do množiny.	s = {1, 2}; s.add(3) # Output: {1, 2, 3}
	remove()	Odstranění prvku (vyvolá chybu, pokud prvek neexistuje).	s = {1, 2}; s.remove(1) # Output: {2}
	discard()	Odstranění prvku (nevyvolá chybu, pokud prvek neexistuje).	s = {1, 2}; s.discard(3) # Output: {1, 2}
	pop()	Odstraní a vrátí náhodný prvek z množiny.	s = {1, 2, 3}; element = s.pop() # Output: 1 (nebo 2 nebo 3)
	union()	Sjednocení dvou množin.	s1 = {1, 2}; s2 = {2, 3}; result = s1.union(s2) # Output: {1, 2, 3}
	intersection()	Průnik dvou množin.	s1 = {1, 2}; s2 = {2, 3}; result = s1.intersection(s2) # Output: {2}
	difference()	Rozdíl dvou množin.	s1 = {1, 2}; s2 = {2, 3}; result = s1.difference(s2) # Output: {1}
	issubset()	Kontrola, zda je množina podmnožinou jiné množiny.	s1 = {1, 2}; s2 = {1, 2, 3}; result = s1.issubset(s2) # Output: True
NoneType	is	Kontrola, zda je hodnota None.	result = (None is None) # Output: True
	is not	Kontrola, zda hodnota není None.	result = (None is not None) # Output: False
bytes	+	Spojení bytových sekvencí.	b1 = b'hello'; b2 = b' world'; result = b1 + b2 # Output: b'hello world'
	*	Opakování bytové sekvence.	b = b'abc'; result = b * 2 # Output: b'abcabc'
	len()	Délka bytové sekvence.	b = b'hello'; result = len(b) # Output: 5
	decode()	Dekódování bytové sekvence do řetězce.	b = b'hello'; result = b.decode('utf-8') # Output: 'hello'
	in	Kontrola, zda prvek je součástí bytové sekvence.	b = b'hello'; result = b'ell' in b # Output: True
bytearray	append()	Přidání prvku do bytového pole.	b = bytearray([1, 2]); b.append(3) # Output: bytearray(b'\x01\x02\x03')
	insert()	Vložení prvku na zadaný index.	b = bytearray([1, 2, 3]); b.insert(1, 4) # Output: bytearray(b'\x01\x04\x02\x03')

	pop()	Odstranění a vrácení posledního prvku (nebo podle indexu).	b = bytearray([1, 2, 3]); result = b.pop() # Output: 3, b: bytearray(b'\x01\x02')
	remove()	Odstranění prvku z bytearray.	b = bytearray([1, 2, 3, 4]); b.remove(3) # Output: bytearray(b'\x01\x02\x04')
	extend()	Přidání více prvků najednou.	b = bytearray([1, 2]); b.extend([3, 4]) # Output: bytearray(b'\x01\x02\x03\x04')
range	len()	Délka rozsahu.	r = range(1, 5); result = len(r) # Output: 4
	in	Kontrola, zda hodnota je v rozsahu.	r = range(1, 5); result = 3 in r # Output: True
	not in	Kontrola, zda hodnota není v rozsahu.	r = range(1, 5); result = 6 not in r # Output: True
	+	Sčítání dvou rozsahů (Python 3.9+).	r1 = range(1, 3); r2 = range(4, 6); result = list(r1 + r2) # Output: [1, 2, 4, 5]
	*	Opakování rozsahu (Python 3.9+).	r = range(1, 3); result = list(r * 2) # Output: [1, 2, 1, 2]