

Bijlage staatsexamen HAVO/VWO 2026

Informatica

College-examen schriftelijk

Bijlage ASCII-tabel

Bijlage Syntax programmeren Python

Bijlage Overzicht SQL-instructies

Bijlage ASCII-tabel

DEC	CHAR	DEC	CHAR	DEC	CHAR	DEC	CHAR
0	NUL	32	[spatie]	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	'	71	G	103	g
8	BS	40	(72	H	104	h
9	HT	41)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	.	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	ETB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[123	{
28	FS	60	<	92	\	124	
29	GS	61	=	93]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	_	127	[del]

Bijlage Syntax programmeren Python

String in variabele

```
my_str = "Hallo Wereld"
print (my_str)
Resultaat: Hallo Wereld
```

Ingebouwde functies

print() toont gewenste informatie op het scherm
int() verandert getal naar geheel getal
float() verandert getal naar decimaal getal
str() een reeks met getal, letters of symbolen erin
len() geeft de lengte van een string, lijst, bibliotheek
sys.exit() beëindigt de uitvoering van het programma

```
str1 = "Welkom bij deze Python-uitleg"
print ("De lengte van deze zin is: ", len (str1), "tekens.")
```

Resultaat: De lengte van deze zin is: 29 tekens.

filter() is te gebruiken bij het uitsluiten van items in veranderbare objecten
zoals lijsten, bibliotheken, enz.

```
leeftijd =[5,12,17,13,24,65,18]
def myFunc(x):
    if x < 18:
        return False
    else:
        return True
volwassen = filter(myFunc, leeftijd)
for x in volwassen:
    print(x)
```

Resultaat: 24
 65
 18

Keuzestructuur IF statement

```
if <voorwaarde 1>:
    <programmeerinstructie(s)>
elif <voorwaarde 2>:
    <programmeerinstructie(s)>
elif <voorwaarde 3>:
    <programmeerinstructie(s)>
else:
    <programmeerinstructie(s)>
```

FOR loop

```
# herhaling basis
for i in range(n):
    loop <programmeerinstructie(s)>

# herhaling in rij objecten
for <variabele> in <rij objecten>:
    <programmeerinstructie(s)>

# herhaling met string
for <variabele> in <string>:
    print(<variabele>)
```

WHILE Loop

```
while True: #altijd
    gebruikers_invoer = input("Geef een getal: ")
    getal = int(gebruikers_invoer)
    print ('Het kwadraat van het getal is: ', getal ** 2)
```

Maken van een functie

```
def optellen(x, y, z):
    a = x+y
    b = x+z
    c = y+z
    print(a,b,c)
optellen (1, 2, 3)
Resultaat: 3 4 5
```

Lijsten

```
mylist = list((2,3,5)) #maak een lijst
print (mylist[0])      #selecteert het eerste onderdeel en toont dat
print (len(mylist))   #toont de lengte (3)
mylist.append(3)      #voegt een element toe achteraan de lijst
mylist.remove(1)     #verwijdert een element van de lijst
```

Andere mogelijke methoden bij het gebruik van een lijst:

clear(), copy(), count(), extend(), index(), insert(), pop(), reverse(), sort()

Bijlage Overzicht SQL-instructies

(Alles tussen [] is niet verplicht)

| betekent 'of'

... betekent '1 of meer'

In te vullen gegevens staan tussen < >

Query's hoeven bij het examen niet te worden afgesloten met ;

Basisstructuur Query

Selecteren

```
SELECT [DISTINCT] <kolomnaam[,...] en/of functies> [geen dubbele rijen]
FROM <tabellen>
    [ WHERE <voorwaarde[n]> ]
    [ ORDER BY <sorteer-kolomme[n]> [ASC|DESC] |rangnummer uit SELECT
]
    [ GROUP BY <groepeer-kolomme[n]> ]
    [ HAVING <groep-voorwaarde[n]> ]
```

Kolom of tabel hernoemen met een alias

```
SELECT <langekolomnaam> AS <kortekolomnaam>
```

```
SELECT <x.kolomnaam[,...]>
FROM <tabelnaam> AS <x[,...]>
```

Tabellen koppelen

```
FROM <tabel1> INNER JOIN <tabel2> ON <tabel1.kolomnaam =
tabel2.kolomnaam>
```

Creëren

```
CREATE TABLE <tabelnaam>
    (<kolomnaam> <gegevenstype> [NOT NULL] [,...]
    [ PRIMARY KEY (<kolomnaam>) [,...] ] )
```

VWO

```
[ FOREIGN KEY (<kolomnaam>) [,(<kolomnaam>) ...] ],
[ REFERENCES <tabelnaam> (<kolomnaam>) ] )
```

Kolommen

```
<kolomnaam> | <tabelnaam>.<kolomnaam> | * | functie
```

Gegevens

```
<gegevenstype> CHAR(1) | DATUM | DEC(t,d) | INT | SMALLINT zijn standaard
SQL-typen
```

Wijzigen

UPDATE <tabelnaam>
SET(<kolomnaam>) = <waarde>
[WHERE <voorwaarde [,...]>]

Invoegen

INSERT INTO <tabelnaam> [(kolomnaam,[...])]
VALUES (<waarde> [...])

Verwijderen

DROP TABLE <tabelnaam> verwijdt hele tabel inclusief structuur
DELETE FROM <tabelnaam>
 [WHERE <voorwaarde [,...]>] verwijdt één, meerdere of alle rijen.

Functies

COUNT(*)	geeft het aantal rijen
SUM(<kolomnaam>)	geeft de som van de waarden
AVG(<kolomnaam>)	geeft het gemiddelde van de waarden
MAX(<kolomnaam>)	geeft de hoogste waarde van de waarden
MIN(<kolomnaam>)	geeft de laagste waarde van de waarden

Vergelijken / logische operatoren

= , < , > , <= , >= , of <> om te vergelijken
BETWEEN waarde1 AND waarde2
 vanaf eerste waarde1 tot en met de tweede waarde2
NOT, AND, of OR om voorwaarden om te keren en te combineren
LIKE ? voor één willekeurige letter
 * voor willekeurig aantal letters
IS NULL om te testen of een veld leeg is
IS NOT NULL om te testen of een veld ingevuld is

Sorteren

ORDER BY <kolomnaam [,...]> ASC | DESC
ORDER BY rangnummer van de kolom uit SELECT

Subquery

WHERE <kolomnaam> IN (<subquery>)
WHERE [NOT] EXISTS (<subquery>)